



Commande des mouvements et de l'équilibre d'un robot humanoïde à roues omnidirectionnelles

Jory Lafaye

► To cite this version:

Jory Lafaye. Commande des mouvements et de l'équilibre d'un robot humanoïde à roues omnidirectionnelles. Automatique / Robotique. Université Grenoble Alpes, 2015. Français. NNT : 2015GREAT052 . tel-01218721

HAL Id: tel-01218721

<https://theses.hal.science/tel-01218721>

Submitted on 21 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Electronique, électrotechnique, automatique et traitement du signal**

Arrêté ministériel : 7 août 2006

Présentée par

Jory Lafaye

Thèse dirigée par **Bernard Brogliato** et
codirigée par **Pierre-Brice Wieber**

préparée au sein du **Laboratoire Inria Grenoble Rhône-Alpes**
dans l'**École Doctorale EEATS**

Commande des mouvements et de l'équilibre d'un robot humanoïde à roues omnidirectionnelles

Thèse soutenue publiquement le **2 juillet 2015**,
devant le jury composé de :

Dr. Nicolas Marchand

Directeur de recherche, GIPSA-lab, Président

Dr. Olivier Stasse

Directeur de recherche, LAAS-CNRS Toulouse, Rapporteur

Dr. Christine Chevallereau

Directeur de recherche, IRCCyN, Rapporteur

Dr. Pierre-Brice Wieber

Chargé de recherche, Inria Grenoble Rhône-Alpes, Membre

Dr. Sébastien Dalibard

Ingénieur, Aldebaran, Membre

Dr. Philippe Fraisse

Professeur, Lirmm, Membre

Dr. Bernard Brogliato

Directeur de recherche, Inria Grenoble Rhône-Alpes, Invité



Résumé

La problématique traitée dans cette thèse concerne la commande et l'équilibre des robots humanoïdes disposant d'une base mobile à roues omnidirectionnelles. Les méthodes développées visent à atteindre de hautes performances dynamiques pour ce type de robot, tout en assurant stabilité et équilibre.

Les robots humanoïdes ont en général un centre de masse relativement haut en comparaison avec leur surface de contact avec le sol. Ainsi, la moindre accélération des corps du robot induit une large variation de la répartition des forces de contact avec le sol. Si celles-ci ne sont pas correctement contrôlées, alors le robot peut tomber. De plus, le robot disposant d'une base mobile à roues, une perturbation peut l'amener aisément à basculer sur deux roues. Enfin, un intérêt particulier a été apporté à la réalisation d'une commande temps-réel implémentée sur le système embarqué du robot. Cela implique principalement des contraintes concernant le temps de calcul de la loi de commande.

Afin de répondre à ces problèmes, deux modèles linéaires du robot ont été réalisés. Le premier permet de modéliser la dynamique du robot lorsque celui-ci possède toutes ses roues en contact avec le sol. Le second permet de modéliser la dynamique du robot lorsque celui-ci bascule sur deux de ses roues. Ces modèles ont été réalisés en prenant en compte la répartition massique du robot. Ainsi, il a été judicieux de le modéliser comme un système à deux masses ponctuelles, pouvant se déplacer sur un plan parallèle au sol. La première correspond au centre de masse de la base mobile, la seconde à celui du reste du robot.

Ces modèles sont ensuite utilisés au sein de deux commandes prédictives, permettant de prendre en compte à chaque instant les contraintes dynamiques ainsi que le comportement du robot dans le futur. La première commande permet de contrôler les déplacements du robot lorsque celui-ci possède toutes ses roues en contact avec le sol, lui assurant de ne pas basculer. La seconde

permet au robot de se rattraper d'une situation où une perturbation l'amène à basculer, afin de ramener toutes ses roues en contact avec le sol.

Aussi, un superviseur disposant d'une machine à état à été réalisé afin de définir quelle loi de commande doit être exécutée à chaque instant. Ce superviseur utilise les capteurs disponibles sur le robot afin d'observer son état de basculement.

Enfin, afin de valider expérimentalement le résultat des développements de cette thèse, une série d'expériences a été présentée, mettant en évidence les différents aspects de la loi de commande. Notamment, des essais ont été réalisés concernant le suivi de trajectoires non physiquement réalisables, le rejet de perturbations appliqués à la base mobile, la stabilisation du robot lors de son basculement, ainsi que la compensation de variations de l'inclinaison du sol.

Abstract

The problem of this thesis concerns the control of the movements and the equilibrium of humanoid robots that have a mobile base with omnidirectional wheels. The developed methods aim to reach high dynamical performances for this type of robot, while ensuring its stability and equilibrium.

Humanoid robots have generally a center of mass relatively high compared to its contact surface with the ground. Therefore, the slightest acceleration of the robot bodies induces a large variation of the distribution of the contact forces with the ground. If they are not properly controlled, the robot can fall. Moreover, the robot having a mobile base with wheels, a disturbance can easily bring it to tilt on two wheels. Finally, a specific interest has been provided about the realization of a real time controller implemented on the embedded system of the robot. This implies some constraints about the computational time of the control law.

In order to answer these problems, two linear models of the robot have been developed. The first allows to modelize the dynamics of the robot when it has all of its wheels in contact with the ground. The second allows to modelize the dynamics of the robot when it tilts on two of its wheels. These models have been developed by taking into account the mass distribution of the robot.

These models have been subsequently used in two predictive control laws, allowing to take into account at every instant the dynamical constraints as well as the future behavior of the robot. The first allows to control the movements of the robot when it has all of its wheels in contact with the ground, preventing it from tilting. The second allows the robot to recover itself in a situation when a disturbance brings it to tilt, in order to bring back all of its wheels in contact with the ground.

Also, a supervisor that has a state machine has been made in order to define which control law has to be executed at each instant. This supervisor uses the available sensors on the robot in order to observe its tilt state.

Finally, in order to validate experimentally the results of the developments of this thesis, a series of experiments has been presented, demonstrating some aspects of the control law. In particular, some tests have been made concerning the tracking of non physically feasible trajectories, the reject of disturbances applied on the mobile base, the stabilisation of the robot during its tilt, and the compensation of the variations of the ground inclination.

Table des matières

I	Introduction	14
I.1	Contexte	14
I.2	Contributions	15
I.3	Etat de l’art	16
I.3.1	Commande et équilibre des robots à roues	17
I.3.1.1	Les robots à une et deux roues	17
I.3.1.2	Les robots à trois roues et plus	19
I.3.1.2.a	Compensation du basculement	19
I.3.1.2.b	Contrôle du mouvement et de l’équilibre	20
I.3.2	Commande et équilibre des robots bipèdes	22
I.3.3	Commande et équilibre des robots compliantes	25
I.3.4	Synthèse	27
I.4	Organisation du document	27
II	Modélisation du système	29
II.1	Choix du modèle et conséquences	29
II.1.1	Objectifs	29
II.1.2	Avantages du modèle	29
II.1.3	Inconvénients du modèle	31
II.2	Modélisation dynamique	32
II.2.1	Problème de complémentarité mixte	32
II.2.1.1	Géométrie du robot	32

II.2.1.2	Hypothèses	32
II.2.1.3	Cinématique directe	34
II.2.1.3.a	Notations	34
II.2.1.3.b	Equations cinématiques	35
II.2.1.3.c	Hypothèse d'angles de basculements faibles	37
II.2.1.4	Énergies cinétiques et potentielles	38
II.2.1.4.a	Contraintes sur le basculement du robot	38
II.2.1.4.b	Formulation des énergies cinétiques et potentielles	40
II.2.1.5	Coordonnées généralisées et Lagrangien	41
II.2.1.6	Application du principe de moindre action	41
II.2.1.6.a	Définitions	41
II.2.1.6.b	Équations de la dynamique	43
II.2.1.6.c	Formulation standard	44
II.2.1.7	Contraintes de complémentarité	46
II.2.1.8	Synthèse	47
II.2.2	Cas où les trois roues sont en contact avec le sol	47
II.2.2.1	Modèle dynamique	47
II.2.2.2	Définition du Centre de Pression	49
II.2.2.3	Principe de conservation du moment angulaire	49
II.2.2.4	Formulation du Centre de Pression et simplifications	50
II.2.3	Cas où le robot bascule sur deux roues	51
II.2.3.1	Modèle dynamique	51
II.2.3.2	Formulation du Centre de Pression	53
II.2.3.3	Simplifications	55
II.3	Modélisation de la dynamique future	56
II.3.1	Nécessité de prédire le futur	56
II.3.2	Choix de la dynamique d'extrapolation	57
II.3.3	Formulation du modèle prédictif	59
II.3.3.1	Cas où les trois roues sont en contact avec le sol	59
II.3.3.2	Cas où le robot bascule sur deux roues	60
II.3.3.2.a	Relations différentielles des variables dynamiques	60

II.3.3.2.b	Formulation de l'équation prédictive de la dynamique . . .	61
II.3.3.2.c	Inversibilité de la dynamique de basculement	61
II.3.4	Synthèse	62
III	Commande par modèle prédictif	63
III.1	Principe	63
III.2	Outil mathématique et contraintes associées	64
III.3	Formulation des problèmes d'optimisations	65
III.3.1	Introduction	65
III.3.2	Lorsque les trois roues sont en contact avec le sol	66
III.3.2.1	Formulation des objectifs	66
III.3.2.1.a	Suivi de trajectoire	67
III.3.2.1.b	Robustesse aux perturbations	67
III.3.2.1.c	Stabilité numérique	68
III.3.2.2	Formulation des contraintes	69
III.3.2.2.a	Contraintes de non-bascullement	69
III.3.2.2.b	Vitesses et accélérations maximales de la base mobile . .	71
III.3.2.2.c	Limites articulaires du corps du robot	72
III.3.2.3	Problème quadratique résultant	74
III.3.3	Lorsque le robot bascule sur deux roues	75
III.3.3.1	Formulation des objectifs	75
III.3.3.1.a	Minimisation de l'angle de basculement	75
III.3.3.1.b	Minimisation de la vitesse angulaire	76
III.3.3.1.c	Stabilité numérique	76
III.3.3.2	Formulation des contraintes	77
III.3.3.2.a	Contraintes de non-pénétration dans le sol	77
III.3.3.2.b	Vitesses et accélérations maximales de la base mobile . .	78
III.3.3.2.c	Limites articulaires du corps du robot	78
III.3.3.3	Problème quadratique résultant	80
III.3.4	Gestion de la transition entre les deux états	80
III.4	Gestion des deux modèles dynamiques exclusifs	81

III.4.1	Choix d'un superviseur et conséquences	81
III.4.2	Fonctionnement du superviseur	82
III.4.2.1	Les états du superviseur	82
III.4.2.2	Les transitions depuis l'état <i>No tilt</i>	83
III.4.2.3	Les transitions depuis l'état <i>Tilt</i>	84
III.4.2.4	Les transitions depuis l'état <i>Landing</i>	84
III.4.3	Fonctionnement de l'estimateur d'impact	85
III.5	Vers une modélisation unifiée des deux dynamiques	86
III.6	Conclusion	87
IV	Mesures et observateurs	88
IV.1	Introduction	88
IV.2	Valeurs à observer et capteurs disponibles	88
IV.3	Méthodes de mesure et conséquences	89
IV.3.1	Observation de l'état du CoM de la base mobile	89
IV.3.1.1	Vitesse du CoM de la base mobile	89
IV.3.1.2	Position du CoM de la base mobile	90
IV.3.1.3	Accélération du CoM de la base mobile	90
IV.3.2	Observation de l'état du CoM du corps du robot	91
IV.3.2.1	Position du CoM du corps du robot	91
IV.3.2.2	Vitesse et accélération du CoM ddu corps du robot	91
IV.3.3	Observation de l'angle de basculement et d'inclinaison du sol	91
V	Résultats et expérimentations	93
V.1	Introduction	93
V.2	Schéma de contrôle en boucle fermée	93
V.2.1	Boucle de commande moteur	93
V.2.2	Boucle de commande principale	94
V.2.3	Filtrages capteur	95
V.3	Expériences	96
V.3.1	Introduction	96
V.3.2	Suivi de trajectoires non réalisables	98

V.3.2.1	Protocole expérimental	98
V.3.2.2	Analyse	98
V.3.3	Suivi de trajectoire en présence de faibles perturbations	100
V.3.3.1	Protocole expérimental	100
V.3.3.2	Analyse	101
V.3.4	Compensation du basculement du robot	102
V.3.4.1	Lorsque le robot est immobile	102
V.3.4.1.a	Protocole expérimental	102
V.3.4.1.b	Analyse	103
V.3.4.2	Lorsque le robot est en mouvement	104
V.3.4.2.a	Protocole expérimental	104
V.3.4.2.b	Analyse	105
V.3.5	Compensation de l'inclinaison du sol	106
V.3.5.1	Protocole expérimental	106
V.3.5.2	Analyse	107
V.4	Conclusion	107
VI	Synthèse	109
VI.1	Contributions	109
VI.1.1	Modélisation	109
VI.1.2	Loi de commande	110
VI.1.3	Résultats	110
VI.2	Perspectives	111
	Bibliographie	112
	Annexes	117
A	Pepper, un robot humanoïde à roues omnidirectionnelles	118
B	MPC-WalkGen, librairie C++ implémentant la commande par modèle prédictif	120

Table des figures

I.1	Présentation de Pepper, robot humanoïde conçu par Aldebaran	15
I.2	Illustration du robot présenté dans l'article [10]	18
I.3	Illustration du robot présenté dans l'article [11] : Un véhicule étroit capable de déplacer latéralement son centre de masse par rapport à la position de ses roues. . .	20
I.4	Illustration du robot présenté dans l'article [7] : Un véhicule étroit capable de déplacer latéralement son centre de masse par rapport à la position de ses roues. . . .	21
I.5	Illustration du robot présenté dans l'article [26] : Un robot articulé à roues.	22
I.6	Illustration du robot présenté dans l'article [23] : Un robot non-articulé à roues. . .	23
I.7	Illusatrion comparative entre HRP2 (à gauche), un robot humanoïde bipède, et Pepper (à droite). Les deux robots disposent d'articulations permettant de modifier leur posture : cela fait d'eux des robots articulés. Ils disposent également d'une morphologie humanoïde, avec des bras, une tête, un torse et des jambes : cela fait d'eux des robots humanoïdes. Enfin, afin de se déplacer, HRP2 possède des pieds alors que Pepper possède des roues.	24
II.1	Représentation globale du modèle dynamique.	30
II.2	représentation dans le repère \mathcal{R}_w des points de contacts avec le sol et des angles de basculement. r correspond au centre géométrique du robot. b_0 et c_0 sont respectivement les positions initiales de la base mobile et du corps du robot. p_r, p_l et p_f correspondent aux points de contacts entre les roues et le sol. ψ_r, ψ_l et ψ_f correspondent aux angles de basculement selon les trois directions de basculement. Enfin, F_r, F_l et F_f correspondent aux forces de contact entre les roues et le sol. . .	33

II.3	Projection dans le plan (O, \vec{x}, \vec{z}) du modèle du robot. Représentation des variables relatives au corps \bar{c} et à l'angle ψ_f . Celles relatives à \bar{b} , ψ_r et ψ_l ne sont pas représentées, mais correspondent au même schéma.	36
III.1	Représentation des contraintes de non-bascullement. D correspond à la contrainte exacte. D' correspond à la contrainte circulaire inscrite dans D , rendant celle-ci invariante par rotation autour de b . D'' correspond à la linéarisation de la contrainte D' par un octogone, en ajoutant une marge de sécurité ζ	70
III.2	Représentation des contraintes cinématiques du corps du robot. H correspond à la contrainte exacte. H' correspond à la contrainte circulaire inscrite dans H , rendant celle-ci invariante par rotation autour de b . H'' correspond à la linéarisation de la contrainte H' par un carré.	73
III.3	Représentation des contraintes cinématiques du corps du robot lorsque celui-ci bascule. H correspond à la contrainte exacte. k^n correspond à la distance entre le point b et le point d'intersection de l'axe (b, \vec{n}_k) et l'enveloppe H	79
III.4	Présentation du changement des états du superviseur au cours du temps.	82
III.5	Schéma de fonctionnement du superviseur. Les boîtes "No tilt", "Tilt" et "Landing" correspondent aux états du superviseur. $\dot{\psi}_i$ correspond à la future vitesse angulaire d'impact estimée et $\dot{\psi}_i$ une limite à celle-ci. $\dot{\psi}$ correspond à la vitesse angulaire courante et $\dot{\psi}$ une limite à celle-ci.	83
V.1	Schéma de contrôle en boucle fermée.	94
V.2	Illustration du filtre de compensation du retard capteur.	96
V.3	Illustration d'un robot Pepper dans un magasin Softbank Mobile au Japon.	97
V.4	Illustration d'un robot Pepper en interaction dans un supermarché Japonais.	98
V.5	Suivi d'une trajectoire non-réalisable avec un jeu de pondérations priorisant le suivi de trajectoire par rapport à la robustesse.	99
V.6	Suivi d'une trajectoire non-réalisable avec un jeu de pondérations priorisant la robustesse par rapport au suivi de trajectoire.	100
V.7	Suivi d'une trajectoire en présence d'une faible perturbation de la base mobile. . .	101
V.8	Compensation d'un basculement du robot en présence d'une faible perturbation. . .	102
V.9	Compensation d'un basculement du robot en présence d'une perturbation moyenne.	103

V.10	Compensation d'un basculement du robot en présence d'une forte perturbation. . .	104
V.11	Compensation d'un basculement du robot en présence lorsque celui-ci est en mouvement	105
V.12	Compensation de l'inclinaison du sol	106
A.1	Présentation de Pepper, robot humanoïde conçu par Aldebaran	119
A.2	Présentation des articulations de Pepper	119

Chapitre I

Introduction

I.1 Contexte

Un des enjeux de la robotique humanoïde a toujours été d'améliorer l'interaction entre les humains et les robots, la rendant plus naturelle et intuitive. Construire des robots à l'apparence humaine permet de créer une forme d'empathie de l'humain vers le robot, facilitant leurs interactions.

D'un point de vue technique, créer et contrôler convenablement des robots bipède à taille humaine est encore du domaine de la recherche. Les difficultés sont multiples et n'ont pas encore été toutes résolues.

Dans l'optique de réaliser un robot social humanoïde à faible coût, Aldebaran, entreprise française en robotique humanoïde, a choisi de concevoir un robot humanoïde non pas bipède, mais possédant une jambe reliée à une base mobile à roues omnidirectionnelles : Pepper [Fig. I.1]. Pepper est destiné entre autres à être vendu aux particuliers. C'est pour cela que la réalisation d'un robot à faible coût est importante. En terme de contrôle de l'équilibre et de la locomotion, cela a un impact principalement sur la présence et la qualité des capteurs et actionneurs.

L'objectif de cette thèse est de permettre la locomotion et l'équilibre d'un robot humanoïde à roues omnidirectionnelles. La plate-forme expérimentale impose certaines contraintes spécifiques. Notamment, le faible coût du robot implique un choix limité de capteurs et d'actionneurs.

Egalement, un intérêt particulier sera porté sur la réalisation d'un algorithme rapide en terme de temps de calcul. En effet, le processeur embarqué sur le robot doit pouvoir résoudre un grand

nombre de tâches, dont la locomotion et l'équilibre, qui n'en sont qu'une petite partie.



FIGURE I.1 – Présentation de Pepper, robot humanoïde conçu par Aldebaran

I.2 Contributions

La contribution principale apportée par cette thèse est la conception d'une loi de commande optimale et prédictive, permettant de contrôler les déplacements d'un robot humanoïde à base mobile omnidirectionnelle. La loi de commande permet également d'assurer le non-basculement du robot lorsqu'il n'est pas perturbé, ainsi que son rétablissement sur ses trois roues lorsqu'une perturbation l'amène à basculer.

La loi de commande permet de prendre en compte un certain nombre de contraintes, comme les limites cinématiques de la base mobile, les contraintes d'unilatéralité des forces de contact ou les limites articulaires des corps du robot.

Cette loi de commande se base sur deux modèles à deux masses-ponctuelles décrivant la dynamique principale du robot. Ces deux modèles permettent de prendre en compte séparément la dynamique du robot sur trois roues et en basculement sur deux roues.

Aussi, un superviseur a été réalisé, permettant de définir quel modèle dynamique utiliser en fonction de l'état de basculement du robot. Ce superviseur permet en outre d'optimiser la distance parcourue du robot lors de la compensation d'un basculement en prédisant l'existence et la vitesse angulaire d'impact de la roue en l'air avec le sol.

Enfin, cette loi de commande a été implémenté en C++ sous la forme d'une librairie nommée *mpc-walkgen* (plus de détails en annexe B). Cette implémentation est utilisée sur tout les robots Pepper de Aldebaran.

Ce travail de thèse a également donné lieu à deux brevets Européens N°14305584.6 et N°14305585.3 ainsi qu'à deux articles publiés pour des conférences internationales [20] [19].

I.3 Etat de l'art

Il n'existe pas de littérature traitant de la commande et de l'équilibre d'un robot humanoïde à base mobile omnidirectionnelle. Cependant, nous pouvons découper les difficultés techniques de ce type de robot en deux sous-parties.

La première est commune à la robotique bipède humanoïde. Ces robots ayant une morphologie humaine, leur centre de masse est généralement haut en comparaison de leur surface de contact avec le sol. Ainsi, la moindre accélération du centre de masse implique une large variation concernant la répartition des forces de contact entre les pieds et le sol.

La seconde sous-partie est commune à la robotique mobile. Concernant les robots disposant de plus de deux roues, ceux-ci ont la possibilité de basculer, ce qui peut conduire à leur chute. Ces robots ont en outre la particularité de pouvoir se trouver dans des situations dynamiques différentes, étant en contact de toute ou partie de leurs roues avec le sol.

Nous allons dans la suite traiter de l'état de l'art de ces différents robots, tout en montrant en quoi le problème posé dans cette thèse est différent de ceux déjà résolus.

I.3.1 Commande et équilibre des robots à roues

I.3.1.1 Les robots à une et deux roues

Les robots à deux roues sont par nature instables, car ils disposent d'une surface de contact avec le sol étroite. Ils sont généralement modélisés par des pendules inversés et les lois de commandes cherchent à positionner le robot autour de son point d'équilibre.

Ces robots sont par nature très différents de Pepper, car ils ne disposent d'aucune configuration d'équilibre stable. De plus, même en situation de basculement, l'objectif de commande de Pepper est non pas d'amener le robot en son point d'équilibre sur deux roues, mais de le faire retomber sur trois roues. Il faut non pas contrôler l'angle que fait le robot avec le sol autour de son point d'équilibre instable, mais plutôt contrôler la dynamique de rotation du robot afin de ramener la roue en l'air sur le sol, qui est loin du point d'équilibre. Il n'est également pas nécessaire de se retrouver en ce point à vitesse nulle, mais plutôt avec une vitesse et une accélération suffisamment faibles pour éviter un rebond.

Voici quelques articles traitant de la commande de ces robots :

L'objectif présenté dans l'article [36] est de maintenir le robot en équilibre, tout en assurant un contrôle en position/vitesse de son déplacement. La modélisation du robot est réalisée par la méthode lagrangienne. Une loi de commande linéarisante, associée à un PID permet de contrôler le robot.

L'objectif présenté dans l'article [38] est de maintenir le robot en équilibre, tout en assurant un contrôle de son déplacement en translation. La modélisation du robot correspond à celle d'un pendule inversé. Une loi de commande adaptative utilisant la logique floue est présentée afin de garantir l'équilibre du robot et le contrôle de son déplacement.

L'objectif présenté dans l'article [10] est de maintenir le robot en équilibre, tout en assurant un contrôle en force de son déplacement [Fig. I.2]. La modélisation du robot est faite à partir des lois de Newton. Une loi de commande par placement de pôle permet d'assurer l'équilibre du robot et le contrôle en force autour de son point d'équilibre.

Concernant les robots à une roue, de nombreuses lois de commande ont été développées :

L'objectif présenté dans l'article [16] est de maintenir le robot en équilibre, tout en assurant un contrôle en force de son déplacement. La modélisation du robot est celle d'un simple pendule inversé. Une loi de commande linéarisante, associée à un PI permet de contrôler les couples moteurs



FIGURE I.2 – Illustration du robot présenté dans l'article [10]

du robot.

L'objectif présenté dans l'article [25] est également de maintenir le robot en équilibre, tout en assurant un contrôle en force de son déplacement. Le robot est modélisé de façon exhaustive par une dynamique rigide corps complet. Une loi de commande LQR permet de contrôler le mouvement longitudinal alors qu'une loi de commande par mode glissants permet de contrôler le basculement latéral.

Enfin, l'objectif présenté dans l'article [48] est également de maintenir le robot en équilibre, tout en assurant un contrôle en force de son déplacement. Le robot est modélisé de façon exhaustive par une dynamique rigide corps complet. Une loi de commande par placement de pôle a été implémentée afin de contrôler les mouvements et l'équilibre du robot.

Dans ces articles, les lois de commandes ne sont pas applicables à Pepper car l'objectif de commande est non pas de stabiliser le robot autour de son point d'équilibre, mais de le faire retomber au sol, en minimisant la vitesse d'impact. De plus, contrairement à Pepper, les robots présentés

n'ont pas la possibilité de déplacer leurs centre de masse par rapport à leur zone de contact de façon directe.

I.3.1.2 Les robots à trois roues et plus

I.3.1.2.a Compensation du basculement

La caractéristique principale des robots à trois roues est qu'ils disposent d'une surface importante de contact avec le sol, contrairement aux robots à deux roues. Cela leur permet d'être stables et de ne pas basculer lorsque les forces de contact des roues avec le sol sont correctement contrôlées.

Il n'existe que peu de littérature concernant le basculement de robots à trois roues et plus. Cependant, il existe un certain nombre d'article traitant de la stabilité des robots manipulateurs à roues [24]. L'objectif de cet article est de permettre à un véhicule doté d'un bras manipulateur de manipuler de lourds objets tout en conservant son équilibre. La conservation de l'équilibre est assurée par des contraintes sur le centre de pression. Une loi de commande optimale a été réalisée afin de contrôler les déplacements du véhicule ainsi que des bras manipulateurs, formulée sous la forme d'un problème d'optimisation quadratique.

Enfin, il existe un certain nombre d'article traitant du cas particulier des *Narrow Tilting Vehicle*, véhicules étroits pouvant basculer sur le côté. Contrairement à Pepper, ces véhicules n'ont pas un déplacement omnidirectionnel.

L'objectif présenté dans l'article [11] est de garantir l'équilibre d'un véhicule étroit à trois roues [Fig. I.3]. Le véhicule est modélisé en utilisant trois modèles, chacun décrivant sa dynamique en fonction du nombre de contact avec le sol (trois roues, en basculement sur la gauche ou sur la droite). Utiliser plusieurs modèles en fonction des états des contacts avec le sol est pertinent dans le cas de robots à roues pouvant basculer. Une approche similaire, utilisant non pas un mais deux corps sera développée dans la thèse afin de modéliser le robot. L'article ne précise pas quel type de loi de commande est utilisée, et se contente de présenter les avantages et inconvénients d'une modélisation utilisant trois modèles.

L'objectif présenté dans l'article [7] est de garantir l'équilibre d'un véhicule étroit à trois roues [Fig. I.4]. Contrairement à [11], le véhicule présenté a été conçu afin d'assurer le contact permanent des trois roues avec le sol, même en situation de basculement de la cabine. L'article présente une



FIGURE I.3 – Illustration du robot présenté dans l'article [11] : Un véhicule étroit capable de déplacer latéralement son centre de masse par rapport à la position de ses roues.

modélisation multi-corps exhaustive du véhicule. Cette modélisation n'est pas applicable dans le cadre de Pepper, notamment à cause des contraintes concernant le temps de calcul de la commande. De plus, la modélisation du robot [Fig. I.4] ne prend pas en compte la possibilité qu'une perturbation l'amène à basculer. Deux lois de commandes simple de type PID permettent de contrôler l'orientation de la cabine par rapport au sol ainsi que le déplacement du véhicule. Ces lois de commandes ne sont pas suffisantes pour Pepper car le rapport entre la hauteur de son centre de masse et la surface de contact avec le sol est bien plus grand que pour le robot présenté dans l'article. Les effets de l'accélération sur la stabilité du robot sont donc bien plus importants.

I.3.1.2.b Contrôle du mouvement et de l'équilibre

Contrairement à la compensation du basculement des robots mobiles à trois roues et plus, il existe un certain nombre d'articles traitant du contrôle de ces robots. Voici quelques-uns des plus pertinents à mettre en relation avec les problèmes apportés par Pepper :

L'objectif présenté dans l'article [45] est d'assurer le déplacement d'un robot mobile à roues omnidirectionnelles. Ce robot étant très bas, il n'a aucun problème d'équilibre. Afin de commander



FIGURE I.4 – Illustration du robot présenté dans l’article [7] : Un véhicule étroit capable de déplacer latéralement son centre de masse par rapport à la position de ses roues.

le déplacement en accélération du robot, il est présenté plusieurs lois de commandes, allant des simples PID aux lois de commandes stochastiques à logique floues.

L’objectif présenté dans l’article [26] est d’assurer le déplacement et l’équilibre d’un robot humanoïde à base mobile omnidirectionnelle [Fig. I.5]. Il est présenté un modèle multi-corps rigide du robot. Une loi de commande de type PID associée à une contrainte sur le Centre de Pression du robot permet de garantir l’équilibre et le déplacement du robot. Le robot [Fig. I.5] est lourd et possède un centre de masse bas. De plus, il n’est que peu articulé, et peu capable de déplacer son centre de masse de façon dynamique. Ainsi, une loi de commande simple de type PID est suffisante pour contrôler son équilibre. Pepper est beaucoup plus léger, et peut se déplacer avec plus d’accélération. Une loi de commande PID n’est pas suffisante pour atteindre ces performances tout en assurant la stabilité du robot.

L’objectif présenté dans l’article [23] est d’assurer le déplacement d’un robot mobile à roues omnidirectionnelles [Fig. I.6]. Il n’est pas traité du problème de l’équilibre de ces robots, mais plutôt du suivi de trajectoire. La loi de commande présentée est une commande optimale prédictive, prenant en compte les contraintes cinématiques de la base mobile. Cette loi de commande est pertinente afin d’obtenir un suivi de trajectoire performant pour des déplacements demandant

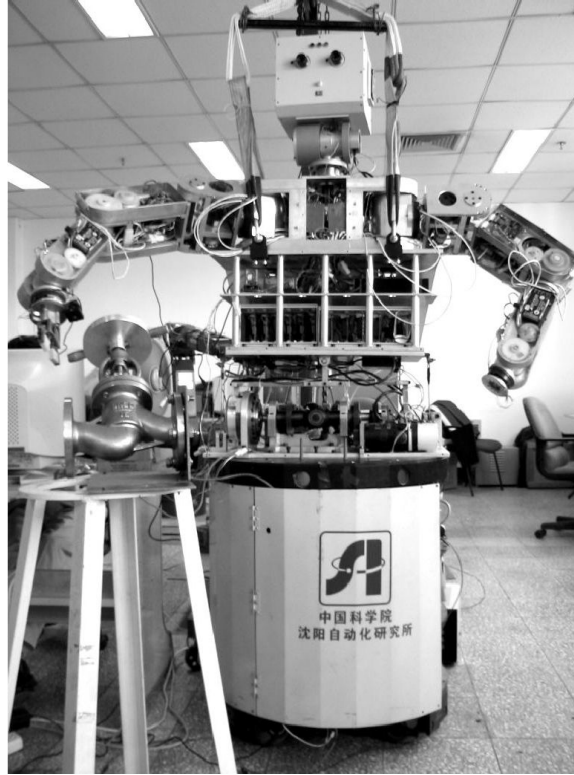


FIGURE I.5 – Illustration du robot présenté dans l’article [26] : Un robot articulé à roues.

beaucoup d’accélération. Une approche similaire sera développée dans la suite de cette thèse, en prenant en compte également l’équilibre du robot, ainsi que sa possibilité de basculer suite à une perturbation.

I.3.2 Commande et équilibre des robots bipèdes

La principale caractéristique des robots bipèdes par rapport aux robots à roues est que pour se mouvoir, ils doivent utiliser les forces de contact entre leurs pieds et le sol. La surface de contact entre le pied et le sol étant fixe, le robot bipède doit enchaîner les pas afin de pouvoir se déplacer.

L’idée principale qui a conduit à cette thèse est le rapprochement que l’on peut faire entre Pepper et un robot bipède : Alors qu’un robot bipède dispose d’une zone de contact avec le sol fixe par

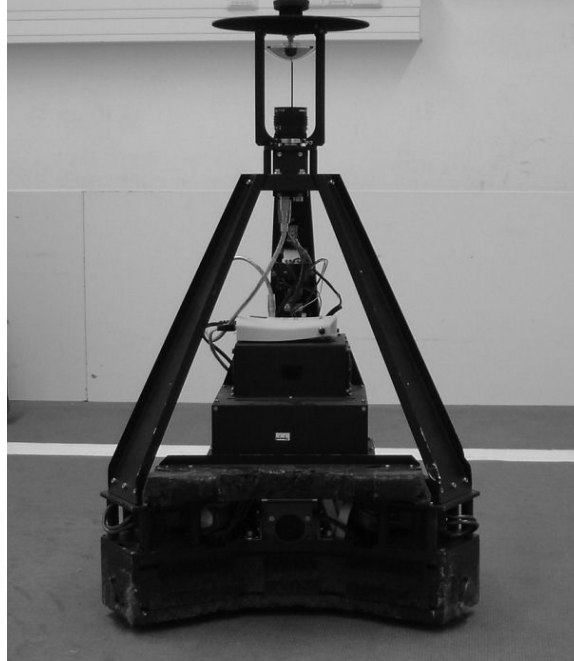


FIGURE I.6 – Illustration du robot présenté dans l'article [23] : Un robot non-articulé à roues.

morceaux (les changements de contacts avec le sol se faisant de manière discrète), Pepper possède une zone de contact avec le sol se déplaçant de manière continue. Les autres caractéristiques de Pepper sont en tout point semblables aux robots humanoïdes bipèdes [Fig. I.7]

Il existe un certain nombre d'articles traitants de la marche des robots bipèdes :

Il est présenté dans l'article [22] une modélisation de la marche des robots bipèdes utilisant un modèle de type pendule inversé. Cette modélisation a servi de base à de nombreux articles traitant de la commande de la marche.

Il est présenté dans l'article [39] un critère permettant de quantifier l'équilibre d'un robot bipède : Le Centre de Pression. Lorsque la surface de contact entre le robot et le sol est défini dans un plan, alors il existe un point, nommé Centre de Pression, qui permet de quantifier si le robot est en mesure de basculer sur le bord de sa surface de contact ou non. Si ce point est strictement inclut dans l'enveloppe convexe de la surface de contact, alors le robot ne peut pas basculer.

Il est présenté dans l'article [21] un algorithme de génération des pas et du déplacement du

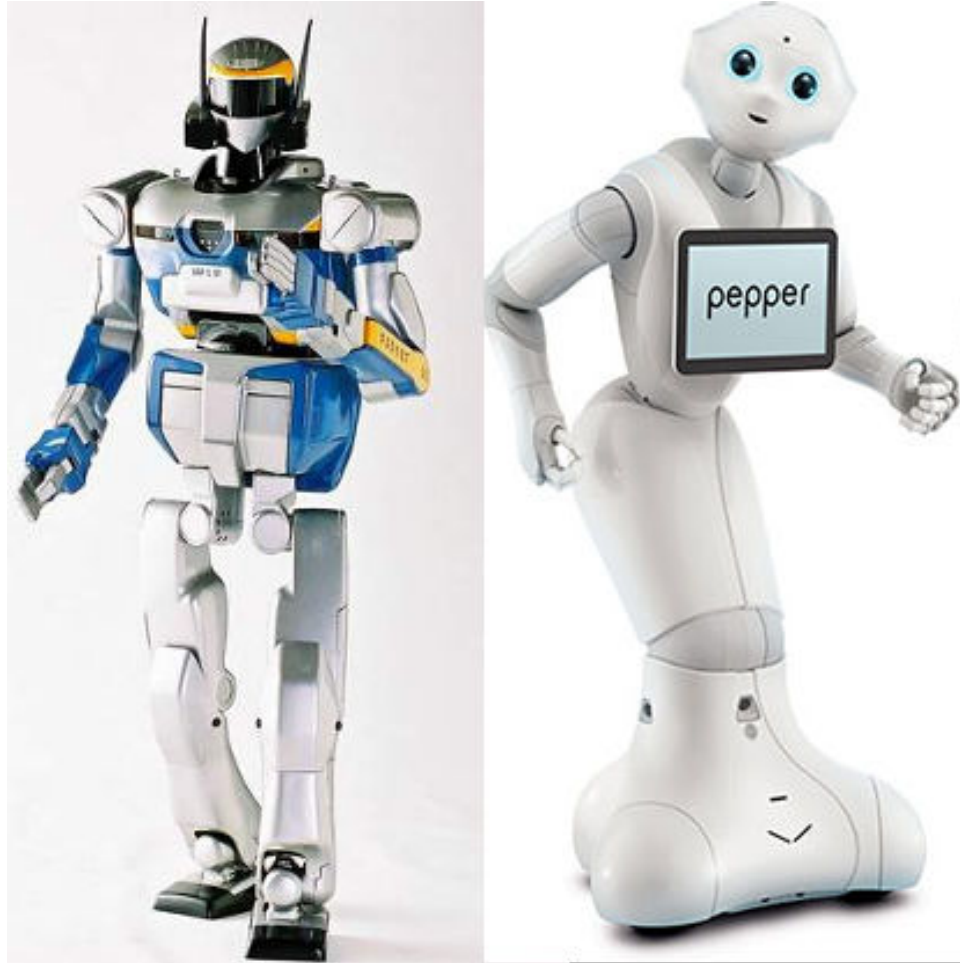


FIGURE I.7 – Illustration comparative entre HRP2 (à gauche), un robot humanoïde bipède, et Pepper (à droite). Les deux robots disposent d'articulations permettant de modifier leur posture : cela fait d'eux des robots articulés. Ils disposent également d'une morphologie humanoïde, avec des bras, une tête, un torse et des jambes : cela fait d'eux des robots humanoïdes. Enfin, afin de se déplacer, HRP2 possède des pieds alors que Pepper possède des roues.

centre de masse du robot afin de garantir une marche stable en l'absence de perturbation. La stabilité de la marche est garantie par l'utilisation du critère du Centre de Pression, qui est contraint à être au centre du pied.

Enfin, il est présenté dans l'article [13] une loi de commande permettant de contrôler à la fois le centre de masse du robot et la position des pas futurs, afin de garantir une marche stable, et un suivi de trajectoire en position/vitesse du centre de masse. La stabilité de la marche est garantie par l'utilisation du critère du Centre de Pression, qui est contraint à être à l'intérieur de l'enveloppe convexe défini par la surface de contact de chaque pied. Cet article nous a servi de base afin de développer la loi de commande présentée dans cette thèse.

Concernant la gestion du basculement des robots bipèdes, on peut trouver nombre de publications utilisant différentes méthodes afin de compenser une perturbation. Certains articles proposent une approche robuste [17]. D'autres articles présentent des approches basées sur une machine à état permettant de définir différentes stratégies de compensation [46, 2]. On trouve également des approches basées sur des méthodes d'apprentissage et de raisonnement probabiliste [40, 18]. Enfin, on trouve des approches basées sur des stratégies de remplacement de pas afin de compenser les perturbations [34, 32, 4]

On peut mettre en parallèle la gestion du basculement de Pepper avec celle des robots bipèdes dans une certaine mesure. Concernant les robots bipèdes, les stratégies classiquement utilisées sont :

- Lorsque la perturbation est faible, utiliser l'articulation de la hanche pour rattraper le robot.
- Lorsque la perturbation est modérée, utiliser en plus l'articulation de la cheville pour rattraper le robot.
- Lorsque la perturbation est forte, réaliser des pas afin de rattraper le robot.

Dans le cas de Pepper, comme nous le verrons dans le chapitre III, les stratégies utilisées sont :

- Lorsque la perturbation est faible, ne rien faire.
- Lorsque la perturbation est modérée ou forte, déplacer la base mobile du robot ainsi que son corps (principalement la hanche ainsi que la cheville) afin de la compenser.

En conclusion, les méthodes de compensation d'une perturbation pouvant emmener le robot à basculer sont relativement similaires. La seule différence concerne le fait qu'un robot bipède doit réaliser des pas pour se rattraper, alors que Pepper utilise ses roues afin de se déplacer.

I.3.3 Commande et équilibre des robots compliant

Au cours des sections précédentes, il a toujours été fait l'hypothèse d'un robot rigide. Pepper a notamment été conçu pour l'être le plus possible, il n'y a donc pas de rapport direct entre les articles traitant de la commande et de l'équilibre des robots compliants et Pepper. Cependant, les robots sont amenés à être de plus en plus compliants dans l'avenir. Dans une optique prospective, il est donc intéressant de se renseigner sur ce qui a été réalisé en terme de commande des robots souples.

Tout d'abord, concernant les bras manipulateurs, on peut trouver diverses stratégies afin de modéliser et contrôler les compliances. L'article [8] présente une modélisation des compliances via un modèle de ressort linéaire et utilise une loi de commande classique de type LQR. L'article [1] présente la modélisation d'un bras manipulateur disposant d'un muscle artificiel en tant qu'actionneur. La modélisation dynamique du muscle est faite en discrétisant la fonction non-linéaire reliant son déplacement et sa force d'actionnement. Un contrôleur PI avec système anti-saturation (*anti windup system*) est utilisé afin de contrôler l'actionneur en force. L'article [43] présente le contrôle d'un robot disposant de tendons élastiques en tant que système de transmission entre les articulations et les moteurs. La modélisation dynamique des tendons est faite à partir de la méthode Lagrangienne en utilisant un modèle de ressort linéaire. Un contrôleur en position utilisant le modèle linéarisé autour des positions articulaires mesurées est présenté par la suite.

Ensuite, concernant la robotique mobile, on peut trouver divers articles traitant de la stabilisation de robots dont les roues sont reliés par des articulations flexibles. L'article [5] présente une architecture de modélisation et de contrôle permettant de commander une table mobile disposant de deux pieds ayant chacun deux roues. Les deux pieds sont reliés par une liaison flexible. L'article [3] présente une loi de commande d'un robot humanoïde à roues disposant d'articulations flexibles. Un critère sur le Centre de Pression associée à une loi de commande robuste permet de garantir la stabilité du robot en présence des flexibilités. L'article [31] présente une loi de commande permettant de contrôler un véhicule à quatre roues séparées au centre par une liaison flexible. L'approche présentée permet la commande d'une combinaison quelconque de modules à deux roues, chacun d'entre eux étant relié par ces liaisons flexibles.

Enfin, concernant la robotique multipède, on peut trouver de nombreux articles traitant de la marche avec des jambes flexibles. L'article [15] présente une loi de commande permettant une marche passive pour un robot avec des jambes flexibles anthropomorphiques. L'article [27] présente une loi de commande robuste permettant de contrôler un robot humanoïde disposant d'arti-

culations flexibles. L'accent a été mis sur l'identification des paramètres d'élasticités permettant de contrôler efficacement la dynamique du robot. L'article [14] présente une loi de commande permettant de contrôler un robot compliant anthropomorphique. L'architecture de commande permet de contrôler à la fois les couples articulaires et l'élasticité de chaque actionneurs. Ces actionneurs imitent le fonctionnement des muscles agonistes et antagonistes des humains. Enfin, l'article [44] présente une loi de commande permettant de contrôler la course d'un robot quadrupède avec des jambes compliantes. Un modèle basé sur une machine à état permet de définir la dynamique du robot en fonction de l'état de la course. Un contrôle robuste en énergie a été réalisé afin de garantir la stabilité du robot.

I.3.4 Synthèse

Au travers de cet état de l'art, nous avons présenté différents travaux en rapport avec le problème de commande posé par Pepper.

Nous avons présenté des articles traitant de l'équilibre des robots mobiles à une et deux roues. Ces lois de commandes ne sont pas pertinentes concernant le robot Pepper car lorsque celui-ci bascule, l'objectif de commande est de faire tomber le robot sur ces trois roues, et non de le conserver en équilibre sur deux roues.

Nous avons également présenté des articles traitant du suivi de trajectoire des robots à plus de deux roues. Ces lois de commandes sont pertinentes lorsque Pepper possède ses trois roues en contact avec le sol. Cependant, les accélérations réalisables par la base mobiles sont très vite limitées par la hauteur du centre de masse de Pepper.

Nous avons ensuite présenté des articles traitant de l'équilibre des robots bipèdes. Nous avons mis en évidence les similarités que Pepper a avec les robots bipèdes, ce qui nous a permis d'utiliser ces travaux comme base pour la commande de Pepper.

I.4 Organisation du document

Au cours du chapitre II, nous présenterons les modèles utilisés pour la commande du robot. Nous décrirons tout d'abord le modèle géométrique, puis nous utiliserons la méthode Lagrangienne afin d'obtenir une modélisation complète du système. Les contraintes de complémentarités apportées par le choix du nombre de contact entre les roues et le sol nous amènerons ensuite à

scinder le modèle en deux sous-parties. Enfin nous étendrons ces modèles dans le futur en utilisant une dynamique d'extrapolation. Ces modèles prédictifs serviront de base à la loi de commande.

Au cours du chapitre III, nous présenterons la loi de commande ainsi que le superviseur. Nous présenterons tout d'abord le formalisme mathématique utilisé afin de résoudre le problème. Ensuite nous présenterons les contrôleurs optimaux associés à chacun des modes dynamiques du système. Enfin nous présenterons le fonctionnement du superviseur.

Au cours du chapitre IV, nous présenterons les différents capteurs utilisés pour le fonctionnement de la loi de commande ainsi que les observateurs utilisés pour les données non directement mesurées.

Au cours du chapitre V, nous réaliserons des expériences sur le robot permettant de mettre en évidence différents aspects de la loi de commande. Notamment, nous présenterons le comportement de la loi de commande lorsque la trajectoire de référence est non-physiquement réalisable. Nous présenterons également les performances de compensation de perturbations amenant le robot à basculer. Enfin nous montrerons les performances de la loi de commande à compenser des variations de l'inclinaison du sol.

Chapitre II

Modélisation du système

II.1 Choix du modèle et conséquences

II.1.1 Objectifs

L'objectif de ce chapitre est de présenter une modélisation dynamique d'un robot humanoïde possédant une base mobile à roues omnidirectionnelles. Ce modèle doit apporter un bon compromis entre fidélité vis à vis du comportement du robot réel et complexité, qui impacte de manière directe le temps de calcul. Notamment, on montre qu'il n'est pas nécessaire de modéliser tout les paramètres du robot : représenter uniquement les dynamiques principales suffit à obtenir un contrôle précis du robot. La section V.2 détaillera les méthodes de compensation des éléments non modélisés.

II.1.2 Avantages du modèle

Le choix se porte sur une modélisation dynamique d'un robot rigide multi-corps. Le modèle présenté [Fig. II.1] comporte deux corps. Le premier est attaché à la base mobile, de masse m_b et de position du centre de masse (CoM) \bar{b} . Le second modélise le corps du robot, de masse m_c et de CoM \bar{c} .

Le choix d'un modèle à deux corps permet de prendre en compte la rotation générale du corps du robot autour de la base mobile. Cette modélisation est pertinente dans le cas où les bras du

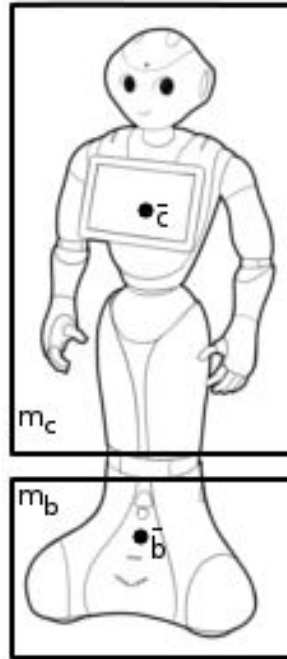


FIGURE II.1 – Représentation globale du modèle dynamique.

robot ne génèrent que peu ou pas de moment angulaire. Dans le cas contraire, nous considérerons que les effets parasites dûs aux mouvements des bras pourront être compensés correctement par le schéma de contrôle en boucle fermée présenté en section V.2. Le choix de ce modèle est également conditionné par la répartition massique de la plate-forme expérimentale : Elle est principalement concentré en deux zones, qui correspondent aux corps choisis : la base mobile et le torse du robot.

Une première approche plus classique basée sur un modèle simple masse de type pendule inversé a été réalisée en premier lieu. Cependant, concernant l'estimation de la position du Centre de Pression, en comparaison d'un un modèle à deux masses, celle-là s'est avérée moins précise de 20% par rapport à une mesure externe. Cela est dû à la répartition massique de la plate-forme expérimentale, qui est concentrée à part égale en deux zones distinctes.

II.1.3 Inconvénients du modèle

Le choix d'un modèle rigide multi-corps implique que les éléments suivants ne seront pas modélisés :

- Les différentes élasticités. Les technologies d'actionnement utilisées sur la plate-forme expérimentale ne comportent pas d'élasticités notables. Le seul élément compliant est un ensemble de deux bandes élastiques attachées à l'articulation du roulis de la hanche permettant au robot de maintenir une posture droite en l'absence de contrôle du moteur. Cet élément est négligeable en terme de dynamique car la raideur associée est très faible.
- Les jeux mécaniques présents sur le robot. Ceux-ci sont présents sur la plate-forme expérimentale, du fait des systèmes de réduction entre les moteurs et les articulations basés sur des engrenages. Les effets dynamiques parasites apportés par le jeu mécanique ne sont pas négligeables. Cependant, il peuvent être compensés de manière suffisamment efficace (plus de détails en section V.2) pour que cela soit transparent du point de vue de la commande présentée dans le chapitre III.
- Les glissements pouvant survenir entre les roues du robot et le sols ne sont pas modélisés. Ceux-ci peuvent être néanmoins handicapants, car le système devient en partie non-observable en présence de glissement (plus de détails en section IV.3.1). Une solution a été apportée en section III.3.2.1 afin de limiter leurs possibilités d'apparition ainsi que leurs impacts sur la dynamique du robot.
- Le nombre de corps choisi pour cette modélisation dynamique est nécessairement plus faible que le nombre de corps réels présents sur le robot, pour des raisons de complexité du modèle. Ainsi, tout les effets dynamiques ne pourront pas être représentés. Le choix du nombre de corps, et de leurs propriétés doit permettre de rendre négligeable les dynamiques non modélisées.

II.2 Modélisation dynamique

II.2.1 Problème de complémentarité mixte

II.2.1.1 Géométrie du robot

On considère le robot modélisé par deux corps \bar{b} et \bar{c} de masse associée m_b et m_c . Ces corps sont en contact avec le sol par l'intermédiaire de trois points p_f , p_r et p_l correspondant aux trois points de contact des roues avec le sol [Fig. II.2]. La roue avant gauche correspond au point p_r , la roue avant droite au point p_l et la roue arrière au point p_f . On associe les indices r , f et l aux différents sens du basculement. r correspond à un basculement vers la droite, f vers l'avant et l vers la gauche. Par exemple, pour un basculement vers la droite, la roue en l'air correspond à la roue avant gauche, qui est nommée p_r .

On considère que le système peut se retrouver dans quatre modes dynamiques différents :

- Le robot ne bascule pas et les trois roues sont en contact avec le sol.
- Le robot est en rotation vers l'avant autour de l'axe défini par les deux roues avant. On note l'angle de rotation ψ_f . La roue arrière est dans ce cas en l'air.
- Le robot est en rotation vers la gauche autour de l'axe défini par la roue avant gauche et la roue arrière. On note l'angle de rotation ψ_l . La roue avant droite est dans ce cas en l'air.
- Le robot est en rotation vers la droite autour de l'axe défini par la roue avant droite et la roue arrière. On note l'angle de rotation ψ_r . La roue avant gauche est dans ce cas en l'air.

Enfin, on considère la répartition de la masse de chaque corps concentrée en un seul point, son CoM. Ainsi, il n'y a aucune inertie de rotation associée à \bar{b} ou \bar{c} .

II.2.1.2 Hypothèses

Dans la suite, nous considérerons un repère galiléen fixe orthonormé direct $\mathcal{R}_w(O, \vec{x}, \vec{y}, \vec{z})$, \vec{x} étant orienté vers l'avant du robot, \vec{y} vers la gauche, et \vec{z} vers le haut. Ce repère est attaché au sol, \vec{x} et \vec{y} inclus dans le plan du sol, et \vec{z} orthogonal au sol. Celui-ci n'est pas forcément horizontal, ce qui implique que le vecteur gravité ne soit pas forcément orienté selon l'axe \vec{z} .

Nous allons définir deux hypothèses conditionnant de façon importante la modélisation du système :

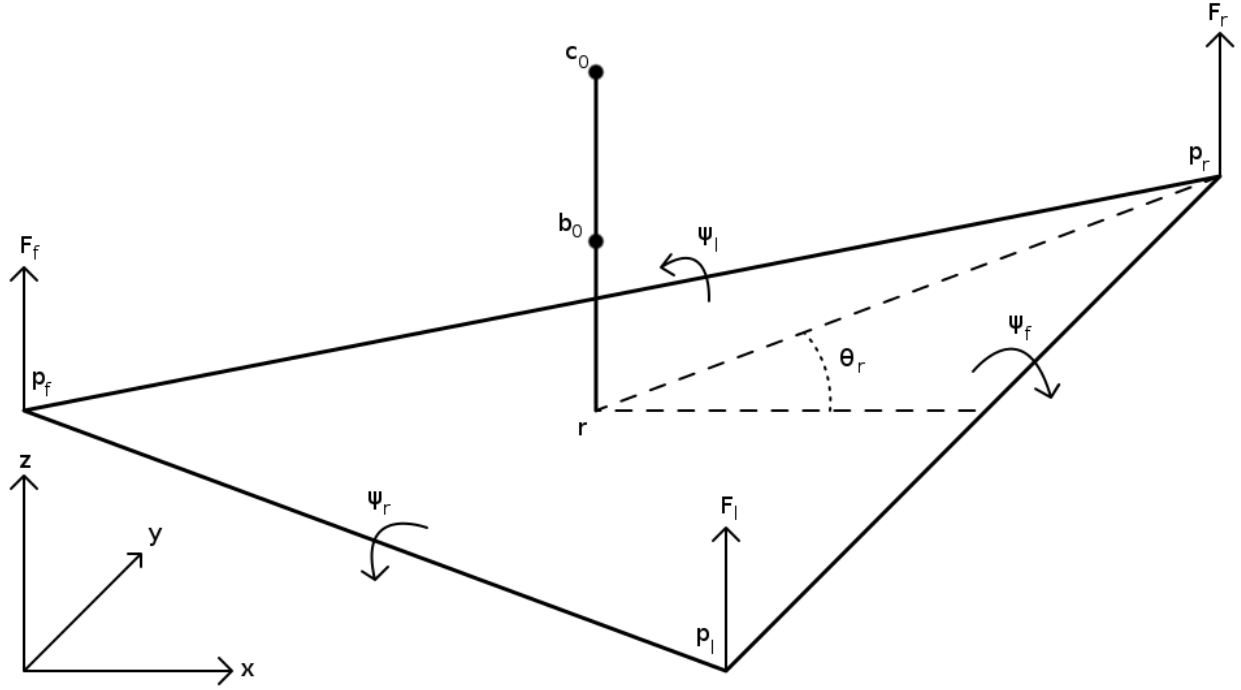


FIGURE II.2 – représentation dans le repère \mathcal{R}_w des points de contacts avec le sol et des angles de basculement. r correspond au centre géométrique du robot. b_0 et c_0 sont respectivement les positions initiales de la base mobile et du corps du robot. p_r , p_l et p_f correspondent aux points de contacts entre les roues et le sol. ψ_r , ψ_l et ψ_f correspondent aux angles de basculement selon les trois directions de basculement. Enfin, F_r , F_l et F_f correspondent aux forces de contact entre les roues et le sol.

- Comme il a été défini en section (II.2.1.1), nous considérons que le robot ne peut se retrouver que sur trois ou deux roues en contact avec le sol. Cela implique que nous ne traitons pas les cas où le robot bascule sur une roue ou est en chute libre.
- La masse du robot étant principalement concentrée sur son axe de rotation selon \vec{z} , nous pouvons négliger les effets dynamiques induits par la rotation du robot. Nous considérons donc l'angle de rotation du robot autour de l'axe \vec{z} comme constant.

II.2.1.3 Cinématique directe

II.2.1.3.a Notations

Premièrement, nous considérons que la position des corps \bar{c} et \bar{b} est obtenue par la composition de trois éléments :

- r , la position générale du centre géométrique de la base mobile projeté sur le sol et exprimée dans le repère \mathcal{R}_w .
- δ_c et δ_b , les longueurs apportées par l'actionnement des moteurs.
- ψ_f , ψ_r et ψ_l , les rotations apportées par le basculement du robot sur chacun de ses cotés.

Afin de pouvoir écrire les équations cinématiques, nous avons besoin de définir différentes grandeurs dépendantes de la géométrie du robot.

Dans la suite de cette section, on note i l'identifiant de la roue i , pouvant prendre les valeurs f , r ou l . $\forall i \in \{f, r, l\}$:

On note θ_i l'angle dans le plan (\vec{x}, \vec{y}) que fait la droite reliant le centre de la base mobile et la roue i avec l'axe \vec{x} [Fig. II.2] :

$$\cos(\theta_i) = \overrightarrow{rp_i} \cdot \vec{x} \quad (\text{II.1})$$

On note \vec{n}_i l'axe défini par la droite reliant le centre de la base mobile r et la roue i , et \vec{t}_i son axe orthogonal inclus dans le plan (\vec{x}, \vec{y}) :

$$\vec{n}_i = \frac{\overrightarrow{r^{xy} p_i^{xy}}}{\left\| \overrightarrow{r^{xy} p_i^{xy}} \right\|} \quad (\text{II.2})$$

$$\vec{t}_i = \vec{z} \times \vec{n}_i \quad (\text{II.3})$$

\mathcal{R}_i représente le repère $(O, \vec{n}_i, \vec{t}_i, \vec{z})$.

On note D_i la distance entre la position p_i de la roue i et la position de la projection des deux autres roues sur l'axe \vec{n}_i (les deux points projetés sont confondus sur l'axe \vec{n}_i car le triangle formé par la base mobile est équilatéral) [Fig. II.3] :

$$D_i = \left\| p_i - p_j \cdot \vec{n}_i \right\| \quad (\text{II.4})$$

avec $j \in \{f, r, l\} \neq i$.

On note d_i la distance dans le plan (\vec{x}, \vec{y}) entre le centre de la base mobile r et la projection des deux autres roues sur l'axe \vec{n}_i [Fig. II.3] :

$$d_i = D_i - \left\| \overrightarrow{r^{xy} p_i^{xy}} \right\| \quad (\text{II.5})$$

On note h_b et h_c les hauteurs initiales de \bar{b} et \bar{c} (lorsque $\delta_c = \delta_b = 0$ et $\psi_f = \psi_r = \psi_l = 0$) [Fig. II.3]. On introduit c_0 et b_0 les positions des CoM des corps \bar{b} et \bar{c} à leurs positions initiales [Fig. II.2] :

$$c_0 = r + h_c \vec{z} \quad (\text{II.6})$$

$$b_0 = r + h_b \vec{z} \quad (\text{II.7})$$

On note l_{b_i} et l_{c_i} les distances entre la position p_i de la roue i et les corps \bar{b} et \bar{c} à leurs positions initiales [Fig. II.3] :

$$l_{b_i} = \left\| \overrightarrow{b_0 p_i} \right\| \quad (\text{II.8})$$

$$l_{c_i} = \left\| \overrightarrow{c_0 p_i} \right\| \quad (\text{II.9})$$

On note ϕ_{i_b} et ϕ_{i_c} les angles que font l'axe \vec{z} avec les droites reliant les positions initiales des corps b_0 , c_0 et la position p_i de la roue i [Fig. II.3] :

$$\cos(\phi_{i_b}) = -\overrightarrow{p_i b_0} \cdot \vec{z} \quad (\text{II.10})$$

$$\cos(\phi_{i_c}) = -\overrightarrow{p_i c_0} \cdot \vec{z} \quad (\text{II.11})$$

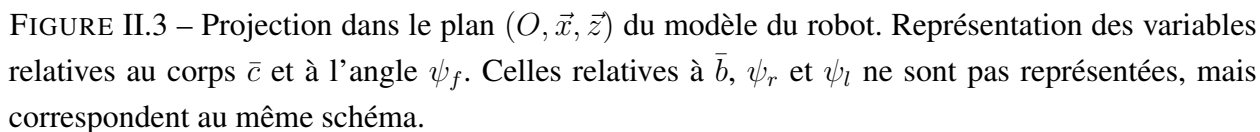
Enfin, on introduit b et c les positions commandées des corps \bar{b} et \bar{c} [Fig. II.3] :

$$b = b_0 \quad (\text{II.12})$$

$$c = c_0 + \delta_c \quad (\text{II.13})$$

II.2.1.3.b Equations cinématiques

Nous pouvons à présent écrire les équations cinématiques des points de contact des roues p_f , p_r et p_l [Fig. II.3]. $\forall i \in \{f, r, l\}$: la position du point de contact p_i peut s'exprimer comme la


$$\begin{cases} p_i^x = b^x + \left(d_i - D_i \cos(\psi_i)\right) \cos(\theta_i) \\ p_i^y = b^y + \left(d_i - D_i \cos(\psi_i)\right) \sin(\theta_i) \\ p_i^z = b^z - h_b + D_i \sin(\psi_i) \end{cases} \quad (\text{II.14})$$

36

On écrit donc, concernant le corps \bar{c} :

$$\begin{cases} \bar{c}^x = c_0^x + \delta^x + \sum_{i \in \{f,r,l\}} \left((d_i + l_{c_i} \sin(\psi_i - \phi_{i_c})) \cos(\theta_i) \right) \\ \bar{c}^y = c_0^y + \delta^y + \sum_{i \in \{f,r,l\}} \left((d_i + l_{c_i} \sin(\psi_i - \phi_{i_c})) \sin(\theta_i) \right) \\ \bar{c}^z = c_0^z + \delta^z + \sum_{i \in \{f,r,l\}} \left(-h_c + l_{c_i} \cos(\psi_i - \phi_{i_c}) \right) \end{cases} \quad (\text{II.15})$$

que l'on peut reformuler en utilisant l'équation (II.13) :

$$\begin{cases} \bar{c}^x = c^x + \sum_{i \in \{f,r,l\}} \left((d_i + l_{c_i} \sin(\psi_i - \phi_{i_c})) \cos(\theta_i) \right) \\ \bar{c}^y = c^y + \sum_{i \in \{f,r,l\}} \left((d_i + l_{c_i} \sin(\psi_i - \phi_{i_c})) \sin(\theta_i) \right) \\ \bar{c}^z = c^z + \sum_{i \in \{f,r,l\}} \left(-h_c + l_{c_i} \cos(\psi_i - \phi_{i_c}) \right) \end{cases} \quad (\text{II.16})$$

et concernant le corps \bar{b} :

$$\begin{cases} \bar{b}^x = b^x + \sum_{i \in \{f,r,l\}} \left((d_i + l_{b_i} \sin(\psi_i - \phi_{i_b})) \cos(\theta_i) \right) \\ \bar{b}^y = b^y + \sum_{i \in \{f,r,l\}} \left((d_i + l_{b_i} \sin(\psi_i - \phi_{i_b})) \sin(\theta_i) \right) \\ \bar{b}^z = b^z + \sum_{i \in \{f,r,l\}} \left(-h_b + l_{b_i} \cos(\psi_i - \phi_{i_b}) \right) \end{cases} \quad (\text{II.17})$$

II.2.1.3.c Hypothèse d'angles de basculements faibles

Dans la suite, afin de simplifier les formulations des équations de la dynamique, nous allons établir dès maintenant une hypothèse : Les angles $\psi_{f,r,b}$ sont considérés proches de 0. Dans le cas où le robot ne bascule pas, cela est exact. Dans le cas où le robot bascule, on suppose que l'angle de basculement reste faible. Nous pouvons donc réécrire la formulation des équations cinétiques des corps \bar{c} et \bar{b} en utilisant une approximation au premier ordre autour de $\psi_f = \psi_r = \psi_l = 0$:

$$\bar{c}^x = \bar{c}^x|_{\psi_{f,r,l}=0} + \sum_{i \in \{f,r,l\}} \left(\left. \frac{\partial \bar{c}^x}{\partial \psi_i} \right|_{\psi_{f,r,l}=0} \psi_i \right) + o(\psi_{f,r,l}^2) \quad (\text{II.18})$$

$$\bar{b}^x = \bar{b}^x|_{\psi_{f,r,l}=0} + \sum_{i \in \{f,r,l\}} \left(\left. \frac{\partial \bar{b}^x}{\partial \psi_i} \right|_{\psi_{f,r,l}=0} \psi_i \right) + o(\psi_{f,r,l}^2) \quad (\text{II.19})$$

Concernant le corps \bar{c} , l'équation (II.16) s'approxime à l'ordre 1 par :

$$\begin{cases} \bar{c}^x \simeq c^x + \sum_{i \in \{f,r,l\}} \left((d_i + l_{c_i} \sin(-\phi_{i_c}) + \psi_i l_{c_i} \cos(-\phi_{i_c})) \cos(\theta_i) \right) \\ \bar{c}^y \simeq c^y + \sum_{i \in \{f,r,l\}} \left((d_i + l_{c_i} \sin(-\phi_{i_c}) + \psi_i l_{c_i} \cos(-\phi_{i_c})) \sin(\theta_i) \right) \\ \bar{c}^z \simeq c^z + \sum_{i \in \{f,r,l\}} \left(-h_c + l_{c_i} \cos(-\phi_{i_c}) - \psi_i l_{c_i} \sin(-\phi_{i_c}) \right) \end{cases} \quad (\text{II.20})$$

que l'on peut simplifier en relevant [Fig. II.3] que $(l_{c_i} \sin(\phi_{i_c}) = d_i)$ et $(l_{c_i} \cos(\phi_{i_c}) = h_c)$:

$$\begin{cases} \bar{c}^x \simeq c^x + \sum_{i \in \{f,r,l\}} \left(\psi_i h_c \cos(\theta_i) \right) \\ \bar{c}^y \simeq c^y + \sum_{i \in \{f,r,l\}} \left(\psi_i h_c \sin(\theta_i) \right) \\ \bar{c}^z \simeq c^z + \sum_{i \in \{f,r,l\}} \left(\psi_i d_i \right) \end{cases} \quad (\text{II.21})$$

et de même concernant le corps \bar{b} :

$$\begin{cases} \bar{b}^x \simeq b^x + \sum_{i \in \{f,r,l\}} \left(\psi_i h_b \cos(\theta_i) \right) \\ \bar{b}^y \simeq b^y + \sum_{i \in \{f,r,l\}} \left(\psi_i h_b \sin(\theta_i) \right) \\ \bar{b}^z \simeq b^z + \sum_{i \in \{f,r,l\}} \left(\psi_i d_i \right) \end{cases} \quad (\text{II.22})$$

Nous aurons également besoin dans la suite des équations approximées des positions des points de contact des roues avec le sol :

$$\begin{cases} p_i^x \simeq b^x + (d_i - D_i) \cos(\theta_i) \\ p_i^y \simeq b^y + (d_i - D_i) \sin(\theta_i) \\ p_i^z \simeq b^z - h_b + D_i \psi_i \end{cases} \quad (\text{II.23})$$

II.2.1.4 Énergies cinétiques et potentielles

II.2.1.4.a Contraintes sur le basculement du robot

Afin d'exprimer les équations du mouvement du robot, nous allons utiliser la méthode Lagrangienne. Celle-ci est pertinente dans notre cas car elle est particulièrement adaptée à la formulation

sans ambiguïtés de systèmes dynamiques soumis à des forces de contraintes, comme les forces de réactions dues au sol sur les roues. Pour cela, il nous faut dans un premier temps écrire les équations des énergies cinétiques T et potentielles V du système :

$$T = \frac{1}{2}m_b(\dot{\bar{b}}^{x^2} + \dot{\bar{b}}^{y^2} + \dot{\bar{b}}^{z^2}) + \frac{1}{2}m_c(\dot{\bar{c}}^{x^2} + \dot{\bar{c}}^{y^2} + \dot{\bar{c}}^{z^2}) \quad (\text{II.24})$$

$$V = m_bg^x\bar{b}^x + m_cg^x\bar{c}^x + m_bg^y\bar{b}^y + m_cg^y\bar{c}^y + m_bg^z\bar{b}^z + m_cg^z\bar{c}^z \quad (\text{II.25})$$

avec g le vecteur gravité. On rappelle que le repère \mathcal{R}_w étant attaché au sol, le vecteur gravité n'est pas forcément orienté selon la direction verticale \vec{z} , car le sol peut ne pas être horizontal.

De plus, l'angle de rotation du robot autour de l'axe \vec{z} étant considéré comme constant, il n'intervient pas dans l'expression de l'énergie cinétique T .

Un seul des angles ψ_f , ψ_r ou ψ_l peut être non-nul à la fois. Cela s'exprime par la contrainte suivante :

$$\forall (i, j) \in \{f, r, l\}, i \neq j, \psi_i\psi_j = 0 \quad (\text{II.26})$$

II.2.1.4.b Formulation des énergies cinétiques et potentielles

En dérivant les équations cinématiques (II.21)(II.22) et en utilisant la contrainte précédente (II.26), nous pouvons exprimer les vitesses des corps \bar{c} et \bar{b} élevées au carré.

$$\dot{\bar{c}}^{x^2} = \dot{c}^{x^2} + h_c^2 \sum_{i \in \{f,r,l\}} \left(\cos(\theta_i)^2 \dot{\psi}_i^2 \right) + 2h_c \dot{c}^x \sum_{i \in \{f,r,l\}} \left(\cos(\theta_i) \dot{\psi}_i \right) \quad (\text{II.27})$$

$$\dot{\bar{c}}^{y^2} = \dot{c}^{y^2} + h_c^2 \sum_{i \in \{f,r,l\}} \left(\sin(\theta_i)^2 \dot{\psi}_i^2 \right) + 2h_c \dot{c}^y \sum_{i \in \{f,r,l\}} \left(\sin(\theta_i) \dot{\psi}_i \right) \quad (\text{II.28})$$

$$\dot{\bar{c}}^{z^2} = \dot{c}^{z^2} + \sum_{i \in \{f,r,l\}} \left(d_i^2 \dot{\psi}_i^2 \right) + 2\dot{c}^z \sum_{i \in \{f,r,l\}} \left(d_i \dot{\psi}_i \right) \quad (\text{II.29})$$

$$\dot{\bar{b}}^{x^2} = \dot{b}^{x^2} + h_b^2 \sum_{i \in \{f,r,l\}} \left(\cos(\theta_i)^2 \dot{\psi}_i^2 \right) + 2h_b \dot{b}^x \sum_{i \in \{f,r,l\}} \left(\cos(\theta_i) \dot{\psi}_i \right) \quad (\text{II.30})$$

$$\dot{\bar{b}}^{y^2} = \dot{b}^{y^2} + h_b^2 \sum_{i \in \{f,r,l\}} \left(\sin(\theta_i)^2 \dot{\psi}_i^2 \right) + 2h_b \dot{b}^y \sum_{i \in \{f,r,l\}} \left(\sin(\theta_i) \dot{\psi}_i \right) \quad (\text{II.31})$$

$$\dot{\bar{b}}^{z^2} = \dot{b}^{z^2} + \sum_{i \in \{f,r,l\}} \left(d_i^2 \dot{\psi}_i^2 \right) + 2\dot{b}^z \sum_{i \in \{f,r,l\}} \left(d_i \dot{\psi}_i \right) \quad (\text{II.32})$$

Nous obtenons donc la formulation des énergies cinétiques et potentielles en fonction des variables du système :

$$\begin{aligned} T = & \frac{1}{2}m_b(\dot{b}^{x^2} + \dot{b}^{y^2} + \dot{b}^{z^2}) + \frac{1}{2}m_c(\dot{c}^{x^2} + \dot{c}^{y^2} + \dot{c}^{z^2}) + \frac{1}{2}(m_b h_b^2 + m_c h_c^2) \sum_{i \in \{f,r,l\}} \left(\dot{\psi}_i^2 \right) \\ & + \frac{1}{2}(m_c + m_b) \sum_{i \in \{f,r,l\}} \left(d_i^2 \dot{\psi}_i^2 \right) + (m_c h_c \dot{c}^z + m_b h_b \dot{b}^z) \sum_{i \in \{f,r,l\}} \left(d_i \dot{\psi}_i \right) \\ & + (m_b h_b \dot{b}^x + m_c h_c \dot{c}^x) \sum_{i \in \{f,r,l\}} \left(\cos(\theta_i) \dot{\psi}_i \right) + (m_b h_b \dot{b}^y + m_c h_c \dot{c}^y) \sum_{i \in \{f,r,l\}} \left(\sin(\theta_i) \dot{\psi}_i \right) \quad (\text{II.33}) \end{aligned}$$

$$\begin{aligned}
V = & m_b g^x b^x + m_c g^x c^x + m_b g^y b^y + m_c g^y c^y + m_b g^z b^z + m_c g^z c^z \\
& + (m_b g^x h_b + m_c g^x h_c) \sum_{i \in \{f,r,l\}} \left(\cos(\theta_i) \psi_i \right) + (m_b g^y h_b + m_c g^y h_c) \sum_{i \in \{f,r,l\}} \left(\sin(\theta_i) \psi_i \right) \\
& + (m_b + m_c) g^z \sum_{i \in \{f,r,l\}} \left(d_i \psi_i \right)
\end{aligned} \tag{II.34}$$

II.2.1.5 Coordonnées généralisées et Lagrangien

L'étape suivante de la méthode Lagrangienne consiste à exprimer l'équation du Lagrangien en fonction des coordonnées généralisées choisies.

Habituellement, le choix des coordonnées généralisées correspondent aux degrés de libertés du système. Afin de faciliter la formulation du modèle, nous allons plutôt utiliser le jeu de coordonnées q suivant, qui possède plus de variables que les degrés de libertés du système :

$$q = \left(b^x \quad b^y \quad b^z \quad c^x \quad c^y \quad c^z \quad \psi_f \quad \psi_r \quad \psi_l \right)^t \tag{II.35}$$

Le choix d'utiliser les trois angles $\psi_{f,r,l}$ pour représenter la rotation du système autour des trois directions possible permettra par la suite d'exprimer plus simplement les équations de la dynamique.

Le Lagrangien L du système est exprimé à partir des énergies cinétiques et potentielles de la façon suivante :

$$L = T - V \tag{II.36}$$

II.2.1.6 Application du principe de moindre action

II.2.1.6.a Définitions

On définit l'action du système comme l'intégrale temporelle du Lagrangien. Le principe de moindre action énonce que, dans le cas d'un système Lagrangien, l'action est stationnaire. Lorsque celui-ci est contraint par des forces de contact, le principe de moindre action permet d'écrire :

$$\forall q_i \in q, \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = J_{u_i}^t(q) u + \sum_{j \in \{f,r,l\}}^{k \in \{x,y,z\}} \left(\frac{\partial p_j^k}{\partial q_i} F_{p_j}^k \right) \tag{II.37}$$

avec u le vecteur des forces d'actionnement et F_{pj}^k la force de contact de la roue $j \in \{f, r, l\}$ selon l'axe $k \in \{x, y, z\}$. $J(q)_u$ correspond à la matrice Jacobienne des points d'applications des forces d'actionnement.

Le système est soumis à deux forces d'actionnement $u = \begin{pmatrix} u_c^{xyz} & u_b^{xy} \end{pmatrix}^t$:

- La première correspond à la force permettant de déplacer le corps du robot par rapport à la base mobile, on la note u_c . Cette force possède des composantes selon les trois directions spatiales et, selon la troisième loi de Newton, s'applique de façon complémentaire sur c et sur b .
- La seconde correspond à la force permettant de déplacer la base mobile sur le sol, on la note u_b . Cette force possède des composantes selon les directions x et y uniquement..

Afin de clarifier l'écriture des équations de la dynamique, on pose :

$$m_{hc} = m_c h_c \quad (\text{II.38})$$

$$m_{hb} = m_b h_b \quad (\text{II.39})$$

$$m_q = m_{hc} h_c + m_{hb} h_b \quad (\text{II.40})$$

$$m_t = m_c + m_b \quad (\text{II.41})$$

$$C_{\theta_i} = \cos(\theta_i) \quad (\text{II.42})$$

$$S_{\theta_i} = \sin(\theta_i) \quad (\text{II.43})$$

II.2.1.6.b Équations de la dynamique

Les éléments de l'équation (II.37) peuvent être calculés à l'aide des équations des énergies (II.33)(II.34) ainsi que des équations cinématiques (II.14) :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = \begin{pmatrix} m_b \ddot{b}^x + m_b h_b \sum_{i \in \{f, r, l\}} \left(C_{\theta_i} \ddot{\psi}_i \right) \\ m_b \ddot{b}^y + m_b h_b \sum_{i \in \{f, r, l\}} \left(S_{\theta_i} \ddot{\psi}_i \right) \\ m_b \ddot{b}^z + m_b h_b \sum_{i \in \{f, r, l\}} \left(d_i \ddot{\psi}_i \right) \\ m_c \ddot{c}^x + m_c h_c \sum_{i \in \{f, r, l\}} \left(C_{\theta_i} \ddot{\psi}_i \right) \\ m_c \ddot{c}^y + m_c h_c \sum_{i \in \{f, r, l\}} \left(S_{\theta_i} \ddot{\psi}_i \right) \\ m_c \ddot{c}^z + m_c h_c \sum_{i \in \{f, r, l\}} \left(d_i \ddot{\psi}_i \right) \\ (m_q + m_t d_f^2) \ddot{\psi}_f + m_{hb} (C_{\theta_f} \ddot{b}^x + S_{\theta_f} \ddot{b}^y - d_f \ddot{b}^z) + m_{hc} (C_{\theta_f} \ddot{c}^x + S_{\theta_f} \ddot{c}^y + d_f \ddot{c}^z) \\ (m_q + m_t d_r^2) \ddot{\psi}_r + m_{hb} (C_{\theta_r} \ddot{b}^x + S_{\theta_r} \ddot{b}^y - d_r \ddot{b}^z) + m_{hc} (C_{\theta_r} \ddot{c}^x + S_{\theta_r} \ddot{c}^y + d_r \ddot{c}^z) \\ (m_q + m_t d_l^2) \ddot{\psi}_l + m_{hb} (C_{\theta_l} \ddot{b}^x + S_{\theta_l} \ddot{b}^y - d_l \ddot{b}^z) + m_{hc} (C_{\theta_l} \ddot{c}^x + S_{\theta_l} \ddot{c}^y + d_l \ddot{c}^z) \end{pmatrix}, \quad (\text{II.44})$$

$$\frac{\partial L}{\partial q} = \begin{pmatrix} m_b g^x \\ m_b g^y \\ m_b g^z \\ m_c g^x \\ m_c g^y \\ m_c g^z \\ (m_{hc} + m_{hb}) C_{\theta_f} g^x + (m_{hc} + m_{hb}) S_{\theta_f} g^y + m_t d_f g^z \\ (m_{hc} + m_{hb}) C_{\theta_r} g^x + (m_{hc} + m_{hb}) S_{\theta_r} g^y + m_t d_r g^z \\ (m_{hc} + m_{hb}) C_{\theta_l} g^x + (m_{hc} + m_{hb}) S_{\theta_l} g^y + m_t d_l g^z \end{pmatrix}, \quad (\text{II.45})$$

$$J_u^t(q) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{II.46})$$

$$\frac{\partial p_{frl}^x}{\partial q} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \frac{\partial p_{frl}^y}{\partial q} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \frac{\partial p_f^z}{\partial q} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ D_f \\ 0 \\ 0 \end{pmatrix}, \frac{\partial p_r^z}{\partial q} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ D_r \\ 0 \end{pmatrix}, \frac{\partial p_l^z}{\partial q} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ D_l \end{pmatrix} \quad (\text{II.47})$$

II.2.1.6.c Formulation standard

Les équations de la dynamique (II.37) peuvent à présent être réécrites en utilisant la formulation standard utilisée en robotique :

$$M\ddot{q} - f(q) = J_u^t(q)u + J_\lambda^t(q)\lambda \quad (\text{II.48})$$

où, en identifiant avec les équations (II.44)(II.45)(II.47)(II.46) :

$$\lambda = \begin{pmatrix} F_f^x & F_r^x & F_l^x & F_f^y & F_r^y & F_l^y & F_f^z & F_r^z & F_l^z \end{pmatrix}^t \quad (\text{II.49})$$

$$u = \begin{pmatrix} u_c^x & u_c^y & u_c^z & u_b^x & u_b^y \end{pmatrix}^t \quad (\text{II.50})$$

$$M = \begin{pmatrix} M_{bc} & M_{c\psi} \\ M_{c\psi}^t & M_\psi \end{pmatrix}, M_\psi = \begin{pmatrix} m_q + m_t d_f^2 & 0 & 0 \\ 0 & m_q + m_t d_r^2 & 0 \\ 0 & 0 & m_q + m_t d_l^2 \end{pmatrix}, \quad (\text{II.51})$$

$$M_{bc} = \begin{pmatrix} m_b & 0 & 0 & 0 & 0 & 0 \\ 0 & m_b & 0 & 0 & 0 & 0 \\ 0 & 0 & m_b & 0 & 0 & 0 \\ 0 & 0 & 0 & m_c & 0 & 0 \\ 0 & 0 & 0 & 0 & m_c & 0 \\ 0 & 0 & 0 & 0 & 0 & m_c \end{pmatrix}, M_{c\psi} = \begin{pmatrix} m_{hb}C_{\theta_f} & m_{hb}C_{\theta_r} & m_{hb}C_{\theta_l} \\ m_{hb}S_{\theta_f} & m_{hb}S_{\theta_r} & m_{hb}S_{\theta_l} \\ m_{hb}d_f & m_{hb}d_r & m_{hb}d_l \\ m_{hc}C_{\theta_f} & m_{hc}C_{\theta_r} & m_{hc}C_{\theta_l} \\ m_{hc}S_{\theta_f} & m_{hc}S_{\theta_r} & m_{hc}S_{\theta_l} \\ m_{hc}d_f & m_{hc}d_r & m_{hc}d_l \end{pmatrix}, \quad (\text{II.52})$$

$$f(q) = \begin{pmatrix} m_b g^x \\ m_b g^y \\ m_b g^z \\ m_c g^x \\ m_c g^y \\ m_c g^z \\ (m_{hc} + m_{hb})C_{\theta_f}g^x + (m_{hc} + m_{hb})S_{\theta_f}g^y + m_t d_f g^z \\ (m_{hc} + m_{hb})C_{\theta_r}g^x + (m_{hc} + m_{hb})S_{\theta_r}g^y + m_t d_r g^z \\ (m_{hc} + m_{hb})C_{\theta_l}g^x + (m_{hc} + m_{hb})S_{\theta_l}g^y + m_t d_l g^z \end{pmatrix} \quad (\text{II.53})$$

$$J_u^t(q) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, J_\lambda^t(q) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D_f & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_l \end{pmatrix} \quad (\text{II.54})$$

II.2.1.7 Contraintes de complémentarité

Afin de compléter la formulation de la dynamique, il reste maintenant à utiliser les *a priori* que l'on dispose sur notre système et formuler les contraintes de complémentarité.

Dans un premier temps, le robot n'a pas la possibilité de pénétrer dans le sol, nous avons donc :

$$\forall i \in \{f, r, l\}, \begin{cases} \psi_i \geq 0 \\ F_i^z \geq 0 \end{cases} \quad (\text{II.55})$$

Aussi, lorsque que le robot bascule sur un des axes $\psi_{f,rl}$, la force résultante sur la roue opposée est nulle :

$$\forall i \in \{f, r, l\}, (\psi_i \neq 0) \Rightarrow (F_i^{xyz} = 0) \quad (\text{II.56})$$

Ensuite, lorsque la force de contact sur une des roues est non-nulle, le robot ne peut pas basculer sur l'axe opposé :

$$\forall i \in \{f, r, l\}, (F_i^{xyz} \neq 0) \Rightarrow (\psi_i = 0) \quad (\text{II.57})$$

Enfin, nous avons défini en section (II.2.1.4.a) que le robot ne peut pas basculer sur plus d'un des trois axes à la fois :

$$\forall (i, j) \in \{f, r, l\}, i \neq j, \psi_i \psi_j = 0 \quad (\text{II.58})$$

Nous pouvons donc exprimer les contraintes de complémentarité pour le système :

$$\forall (i, j) \in \{f, r, l\}, i \neq j, \begin{cases} 0 \leq \psi_i \perp \psi_j \geq 0 \\ 0 \leq \psi_i \perp F_i^z \geq 0 \\ \psi_i \perp F_i^x \\ \psi_i \perp F_i^y \end{cases} \quad (\text{II.59})$$

II.2.1.8 Synthèse

Dans cette section, nous avons dans un premier temps défini les équations cinématiques associées au modèle du robot. Cela nous a permis d'établir une formulation Lagrangienne de la dynamique, puis de l'identifier à la formulation standard des systèmes mécaniques en robotique. Enfin, les *a priori* que nous connaissons concernant le système dynamique nous ont permis d'établir des contraintes de complémentarité sur celui-ci.

Sans ces contraintes, il aurait été possible de résoudre analytiquement les équations temporelles du mouvement. Cependant, la présence de celles-ci impose une résolution particulière. Une première approche, qui sera développée dans les sections suivantes, est d'énoncer d'autres *a priori* afin de fixer le problème de complémentarité (par exemple, en considérant que le robot n'est pas en possibilité de basculer, ou est en état de basculement sur un axe).

Ces approches ne permettent cependant pas de résoudre le problème complet. Nous détaillerons dans la section (III.5) une méthode de résolution du système complet (II.48)(II.59).

II.2.2 Cas où les trois roues sont en contact avec le sol

II.2.2.1 Modèle dynamique

Lorsque le robot est en situation nominale, ses trois roues sont en contact avec le sol. Les mouvements contrôlés du robot ne doivent également pas le faire basculer sur deux roues. Il est donc pertinent de modéliser le système dynamique dans le cas où les trois roues sont en contact avec le sol, car en l'absence de forte perturbations, le contrôleur développé en section (III.3.2) assure cette hypothèse.

Définir le fait que le robot est en contact avec le sol avec ses trois roues permet de résoudre le

problème de complémentarité de la façon suivante :

$$\psi_f = \psi_r = \psi_l = 0 \quad (\text{II.60})$$

Ainsi, le jeu de variables q devient :

$$q = \begin{pmatrix} b^x & b^y & b^z & c^x & c^y & c^z \end{pmatrix}^t \quad (\text{II.61})$$

et l'on peut écrire le modèle dynamique correspondant :

$$M\ddot{q} - f(q) = J_u^t(q)u + J_\lambda^t(q)\lambda \quad (\text{II.62})$$

$$F_f^z \geq 0, \quad F_r^z \geq 0, \quad F_l^z \geq 0 \quad (\text{II.63})$$

avec :

$$\lambda = \begin{pmatrix} F_f^x & F_r^x & F_l^x & F_f^y & F_r^y & F_l^y & F_f^z & F_r^z & F_l^z \end{pmatrix}^t \quad (\text{II.64})$$

$$u = \begin{pmatrix} u_c^x & u_c^y & u_c^z & u_b^x & u_b^y \end{pmatrix}^t \quad (\text{II.65})$$

$$M = \begin{pmatrix} m_b & 0 & 0 & 0 & 0 & 0 \\ 0 & m_b & 0 & 0 & 0 & 0 \\ 0 & 0 & m_b & 0 & 0 & 0 \\ 0 & 0 & 0 & m_c & 0 & 0 \\ 0 & 0 & 0 & 0 & m_c & 0 \\ 0 & 0 & 0 & 0 & 0 & m_c \end{pmatrix}, \quad (\text{II.66})$$

$$f(q) = \begin{pmatrix} m_b g^x \\ m_b g^y \\ m_b g^z \\ m_c g^x \\ m_c g^y \\ m_c g^z \end{pmatrix}, \quad J_u^t(q) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}, \quad J_\lambda^t(q) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{II.67})$$

II.2.2.2 Définition du Centre de Pression

Dans le cas d'un système dont les positions des forces de contact sont définies sur un plan, il est possible de définir une grandeur nommée Centre de Pression (CoP) d^{xy} . Le CoP correspond au point dans le plan où le moment angulaire de la résultante des forces de contact est nul. La propriété essentielle du CoP est que celui-ci est toujours inclus à l'intérieur du polygone de l'enveloppe convexe définie par la position des forces de contact. Cela est dû aux contraintes de non-pénétration dans le sol des forces de contact (II.63). Lorsque le CoP est strictement à l'intérieur de ce polygone de support, le robot ne peut pas basculer sur deux roues.

Ainsi, l'utilisation du CoP permet de manipuler de façon pertinente la somme des forces de contact afin de permettre au robot de ne jamais basculer de lui-même. Son expression est la suivante :

$$d^x = \frac{\sum_{i \in \{f, r, l\}} (p_i^{yz} \times F_i^{yz})}{\sum_{i \in \{f, r, l\}} (F_i^z)} \quad (\text{II.68})$$

$$d^y = \frac{\sum_{i \in \{f, r, l\}} (p_i^{xz} \times F_i^{xz})}{\sum_{i \in \{f, r, l\}} (F_i^z)} \quad (\text{II.69})$$

$$(\text{II.70})$$

II.2.2.3 Principe de conservation du moment angulaire

Le modèle de notre robot n'est pas un système fermé. Il n'y a donc pas de conservation du moment angulaire : le lagrangien n'est pas invariant par rotation. Cependant, notre système est soumis à deux types de forces différentes : La première est due à la gravité, et dérive donc d'un potentiel, les secondes sont dues à des efforts de contacts, qui sont des contraintes concernant le système. Il est possible d'exprimer le principe de conservation du moment angulaire dans ce cas là.

Soit ν^i ($i \in \{x, y\}$) un vecteur représentant un déplacement infinitésimal et orthogonal à q dans

le plan de vecteur normal \vec{i} :

$$\nu^x = \begin{pmatrix} 0 \\ -b^z \\ b^y \\ 0 \\ -c^z \\ c^y \end{pmatrix}, \nu^y = \begin{pmatrix} -b^z \\ 0 \\ b^x \\ -c^z \\ 0 \\ c^x \end{pmatrix} \quad (\text{II.71})$$

Le principe de conservation du moment angulaire s'exprime :

$$\nu^x \cdot \left(\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \right) = \sum_{i \in \{f, r, l\}} \left(p_i^{yz} \times F_i^{yz} \right) \quad (\text{II.72})$$

$$\nu^y \cdot \left(\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} \right) = \sum_{i \in \{f, r, l\}} \left(p_i^{xz} \times F_i^{xz} \right) \quad (\text{II.73})$$

En utilisant l'équation du modèle dynamique (II.62), on peut donc écrire ce principe autour des axes \vec{y} et \vec{x} :

$$m_b(\ddot{b}^z - g^z)b^x - m_b(\ddot{b}^x - g^x)b^z + m_c(\ddot{c}^z - g^z)c^x - m_c(\ddot{c}^x - g^x)c^z = \sum_{i \in \{f, r, l\}} \left(p_i^x F_i^z - p_i^z F_i^x \right) \quad (\text{II.74})$$

$$m_b(\ddot{b}^z - g^z)b^y - m_b(\ddot{b}^y - g^y)b^z + m_c(\ddot{c}^z - g^z)c^y - m_c(\ddot{c}^y - g^y)c^z = \sum_{i \in \{f, r, l\}} \left(p_i^y F_i^z - p_i^z F_i^y \right) \quad (\text{II.75})$$

II.2.2.4 Formulation du Centre de Pression et simplifications

On rappelle les équations du mouvement sur l'axe \vec{z} (II.62) :

$$m_b(\ddot{b}^z - g^z) + m_c(\ddot{c}^z - g^z) = \sum_{i \in \{f, r, l\}} \left(F_i^z \right) \quad (\text{II.76})$$

En utilisant les équations résultantes du principe de conservation du moment angulaire autour des axes \vec{y} et \vec{x} (II.74)(II.75) ainsi que l'équation (II.76), on peut formuler le CoP de la façon

suivante :

$$d^{xy} = \frac{m_b(\ddot{b}^z - g^z)b^{xy} - m_b(\ddot{b}^{xy} - g^{xy})b^z + m_c(\ddot{c}^z - g^z)c^{xy} - m_c(\ddot{c}^{xy} - g^{xy})c^z}{m_b(\ddot{b}^z - g^z) + m_c(\ddot{c}^z - g^z)} \quad (\text{II.77})$$

Notre objectif va être maintenant de linéariser l'équation (II.77) par rapport aux variables commandées c^{xy} et b^{xy} . Cela va nous permettre d'utiliser un contrôleur basé sur un système linéaire, ce qui est généralement beaucoup plus efficace en terme de temps de calcul qu'un contrôleur basé sur un système non-linéaire.

On peut dans un premier temps noter que b^z est constant à la hauteur h_b , car $b = b_0$. Ensuite, On choisit de négliger les déplacements verticaux. c^z est donc supposé constant à la hauteur h_c .

En utilisant ces éléments, l'équation de la dynamique (II.77) se réécrit :

$$d^{xy} = \frac{m_b g^z b^{xy} + m_b \ddot{b}^{xy} h_b + m_c g^z c^{xy} + m_c \ddot{c}^{xy} h_c - (m_b h_b + m_c h_c) g^{xy}}{(m_b + m_c) g^z} \quad (\text{II.78})$$

Enfin, les trois contraintes (II.63) impliquent que d^{xy} est à l'intérieur du triangle défini par les trois points de contacts :

$$d^{xy} \times (p_r^{xy} - p_f^{xy}) \geq 0 \quad (\text{II.79})$$

$$d^{xy} \times (p_l^{xy} - p_r^{xy}) \geq 0 \quad (\text{II.80})$$

$$d^{xy} \times (p_f^{xy} - p_l^{xy}) \geq 0 \quad (\text{II.81})$$

II.2.3 Cas où le robot bascule sur deux roues

II.2.3.1 Modèle dynamique

Lorsque le robot est soumis à des perturbations suffisamment fortes, le CoP peut atteindre le bord du polygone de support, et faire basculer le robot sur deux roues. Il est donc important de considérer cet état. Le contrôleur développé en section (III.3.3) permet au robot de contrôler son mouvement sur deux roues afin de le ramener à terme les trois roues en contact avec le sol.

Soit $k \in \{f, r, l\}$ l'indice correspondant à la direction de basculement. Définir le fait que le robot bascule dans cette direction permet de résoudre le problème de complémentarité de la façon suivante :

$$\forall (i, j) \in \{f, r, l\}, i \neq j \neq k, \begin{cases} \psi_i = \psi_j = 0 \\ F_k^{xyz} = 0 \end{cases} \quad (\text{II.82})$$

On définit le jeu de variables q' suivant :

$$q' = \left(b^x \quad b^y \quad b^z \quad c^x \quad c^y \quad c^z \quad \psi_k \right)^t \quad (\text{II.83})$$

et l'on peut écrire le modèle dynamique correspondant :

$$M' \ddot{q}' - f(q') = J'_u{}^t(q')u + J'_{\lambda'}{}^t(q')\lambda' \quad (\text{II.84})$$

$$F_i^z \geq 0, \quad F_j^z \geq 0 \quad (\text{II.85})$$

avec :

$$\lambda' = \begin{pmatrix} F_i^x & F_j^x & F_i^y & F_j^y & F_i^z & F_j^z \end{pmatrix}^t \quad (\text{II.86})$$

$$u = \begin{pmatrix} u_c^x & u_c^y & u_c^z & u_b^x & u_b^y \end{pmatrix}^t \quad (\text{II.87})$$

$$M' = \begin{pmatrix} m_b & 0 & 0 & 0 & 0 & 0 & m_{hb}C_{\theta_k} \\ 0 & m_b & 0 & 0 & 0 & 0 & m_{hb}S_{\theta_k} \\ 0 & 0 & m_b & 0 & 0 & 0 & m_{hb}d_k \\ 0 & 0 & 0 & m_c & 0 & 0 & m_{hc}C_{\theta_k} \\ 0 & 0 & 0 & 0 & m_c & 0 & m_{hc}S_{\theta_k} \\ 0 & 0 & 0 & 0 & 0 & m_c & m_{hc}d_k \\ m_{hb}C_{\theta_k} & m_{hb}S_{\theta_k} & m_{hb}d_k & m_{hc}C_{\theta_k} & m_{hc}S_{\theta_k} & m_{hc}d_k & m_q + m_t d_k^2 \end{pmatrix}, \quad (\text{II.88})$$

$$f(q') = \begin{pmatrix} m_b g^x \\ m_b g^y \\ m_b g^z \\ m_c g^x \\ m_c g^y \\ m_c g^z \\ (m_{hc} + m_{hb}) \cos(\theta_k) g^x + (m_{hc} + m_{hb}) \sin(\theta_k) g^y + m_t d_k g^z \end{pmatrix}, \quad (\text{II.89})$$

$$J_u'^t(q') = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, J_{\lambda'}'^t(q') = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{II.90})$$

II.2.3.2 Formulation du Centre de Pression

De manière similaire à ce qui a été fait en section (II.2.2), nous allons pouvoir exprimer l'équation du CoP en fonction des variables du système.

Le principe de conservation du moment angulaire autour des axes y et x nous donne :

$$\begin{aligned}
& \left(m_b(\ddot{b}^z - g^z) + m_b h_b d_k \ddot{\psi}_k \right) \left(b^x + h_b \cos(\theta_k) \psi_k \right) \\
& - \left(m_b(\ddot{b}^x - g^x) + m_b h_b \cos(\theta_k) \ddot{\psi}_k \right) \left(b^z + d_k \psi_k \right) \\
& + \left(m_c(\ddot{c}^z - g^z) + m_c h_c d_k \ddot{\psi}_k \right) \left(c^x + h_c \cos(\theta_k) \psi_k \right) \\
& - \left(m_c(\ddot{c}^x - g^x) + m_c h_c \cos(\theta_k) \ddot{\psi}_k \right) \left(c^z + d_k \psi_k \right) \\
& = \sum_{i \in \{f, r, l\}, i \neq k} \left(p_i^x F_i^z - p_i^z F_i^x \right)
\end{aligned} \tag{II.91}$$

$$\begin{aligned}
& \left(m_b(\ddot{b}^z - g^z) + m_b h_b d_k \ddot{\psi}_k \right) \left(b^y + h_b \sin(\theta_k) \psi_k \right) \\
& - \left(m_b(\ddot{b}^y - g^y) + m_b h_b \sin(\theta_k) \ddot{\psi}_k \right) \left(b^z + d_k \psi_k \right) \\
& + \left(m_c(\ddot{c}^z - g^z) + m_c h_c d_k \ddot{\psi}_k \right) \left(c^y + h_c \sin(\theta_k) \psi_k \right) \\
& - \left(m_c(\ddot{c}^y - g^y) + m_c h_c \sin(\theta_k) \ddot{\psi}_k \right) \left(c^z + d_k \psi_k \right) \\
& = \sum_{i \in \{f, r, l\}, i \neq k} \left(p_i^y F_i^z - p_i^z F_i^y \right)
\end{aligned} \tag{II.92}$$

L'équation du mouvement sur l'axe \vec{z} s'écrit :

$$m_b(\ddot{b}^z - g^z) + m_b h_b d_k \ddot{\psi}_k + m_c(\ddot{c}^z - g^z) + m_c h_c d_k \ddot{\psi}_k = \sum_{i \in \{f, r, l\}, i \neq k} \left(F_i^z \right) \tag{II.93}$$

En utilisant les équations précédentes (II.91)(II.92)(II.93), on peut formuler le CoP de la façon suivante :

$$\begin{aligned}
d^x = & \frac{m_b \left(\ddot{b}^z - g^z + h_b d_i \ddot{\psi}_k \right) \left(b^x + h_b \cos(\theta_i) \psi_k \right) - m_b \left(\ddot{b}^x - g^x + h_b \cos(\theta_i) \ddot{\psi}_k \right) \left(b^z + d_k \psi_k \right)}{m_b(\ddot{b}^z - g^z) + m_c(\ddot{c}^z - g^z) + (m_b h_b + m_c h_c) d_i \ddot{\psi}_k} \\
& + \frac{m_c \left(\ddot{c}^z - g^z + h_c d_i \ddot{\psi}_k \right) \left(c^x + h_c \cos(\theta_i) \psi_k \right) - m_c \left(\ddot{c}^x - g^x + h_c \cos(\theta_i) \ddot{\psi}_k \right) \left(c^z + d_k \psi_k \right)}{m_b(\ddot{b}^z - g^z) + m_c(\ddot{c}^z - g^z) + (m_b h_b + m_c h_c) d_i \ddot{\psi}_k}
\end{aligned} \tag{II.94}$$

$$\begin{aligned}
d^y = & \frac{m_b \left(\ddot{b}^z - g^z + h_b d_i \ddot{\psi}_k \right) \left(b^y + h_b \sin(\theta_i) \psi_k \right) - m_b \left(\ddot{b}^y - g^y + h_b \sin(\theta_i) \ddot{\psi}_k \right) \left(b^z + d_k \psi_k \right)}{m_b (\ddot{b}^z - g^z) + m_c (\ddot{c}^z - g^z) + (m_b h_b + m_c h_c) d_i \ddot{\psi}_k} \\
& + \frac{m_c \left(\ddot{c}^z - g^z + h_c d_i \ddot{\psi}_k \right) \left(c^y + h_b \sin(\theta_i) \psi_k \right) - m_c \left(\ddot{c}^y - g^y + h_c \sin(\theta_i) \ddot{\psi}_k \right) \left(c^z + d_k \psi_k \right)}{m_b (\ddot{b}^z - g^z) + m_c (\ddot{c}^z - g^z) + (m_b h_b + m_c h_c) d_i \ddot{\psi}_k}
\end{aligned} \tag{II.95}$$

II.2.3.3 Simplifications

Les variables commandées sont c^{xy} et b^{xy} . Nous allons considérer les même simplifications que celles de la section (II.2.2) : b^z est constant à la hauteur h_b et on choisit de négliger les déplacements selon l'axe \vec{z} . c^z est donc supposé constant à la hauteur h_c .

Nous allons maintenant considérer que $d_k = 0$. Cela correspond à négliger le mouvement vertical apporté par le basculement. La distance d_k sur le plan (\vec{x}, \vec{y}) du CoM de la base mobile par rapport à la droite de contact étant faible devant sa hauteur h_b , on peut considérer cette approximation valable.

Enfin, les contraintes de non-pénétration dans le sol impliquent que le CoP soit positionné sur la droite reliant les deux roues en contact avec le sol. Nous allons le contraindre à être au centre, afin d'éviter de s'approcher des situations où le robot pourrait basculer sur une seule roue. Cela correspond à l'équation suivante :

$$d^x = b^x - d_k \cos(\theta_k) \tag{II.96}$$

$$d^y = b^y - d_k \sin(\theta_k) \tag{II.97}$$

et en utilisant l'approximation formulée précédemment ($d_k = 0$) :

$$d^x = b^x \tag{II.98}$$

$$d^y = b^y \tag{II.99}$$

En utilisant les équations (II.98)(II.99) , on peut donc faire disparaître le CoP des équations (II.94)(II.95) et exprimer la relation entre la dynamique de basculement $(\psi_k, \dot{\psi}_k, \ddot{\psi}_k)$ et les variables

commandées :

$$\begin{aligned}
-(m_b h_b + m_c h_c) g^z \cos(\theta_i) \psi_k - (m_b h_b^2 + m_c h_c^2) \cos(\theta_i) \ddot{\psi}_k &= m_b h_b \ddot{b}^x + m_c h_c \ddot{c}^x \\
&- (m_b h_b + m_c h_c) g^x + m_c g^z (c^x - b^x)
\end{aligned}
\tag{II.100}$$

$$\begin{aligned}
-(m_b h_b + m_c h_c) g^z \sin(\theta_i) \psi_k - (m_b h_b^2 + m_c h_c^2) \sin(\theta_i) \ddot{\psi}_k &= m_b h_b \ddot{b}^y + m_c h_c \ddot{c}^y \\
&- (m_b h_b + m_c h_c) g^y + m_c g^z (c^y - b^y)
\end{aligned}
\tag{II.101}$$

Dans le repère $\mathcal{R}_k(O, \vec{n}_k, \vec{t}_k, \vec{z})$, on peut combiner les deux équations précédentes (II.100)(II.101). On multiplie l'équation (II.100) par $\cos(\theta_k)$ puis on additionne le résultat à la multiplication de l'équation (II.101) par $\sin(\theta_k)$. On obtient alors l'équation suivante :

$$\begin{aligned}
-(m_b h_b + m_c h_c) g^z \psi_k - (m_b h_b^2 + m_c h_c^2) \ddot{\psi}_k &= m_b h_b \ddot{b}^{n_k} + m_c h_c \ddot{c}^{n_k} + m_c g^z (c^{n_k} - b^{n_k}) \\
&- (m_b h_b + m_c h_c) (\cos(\theta_k) g^x + \sin(\theta_k) g^y)
\end{aligned}
\tag{II.102}$$

L'équation (II.102) représente finalement la relation entre l'évolution de l'angle de basculement et les variables commandées $c^{n_k t_k}$ et $b^{n_k t_k}$.

II.3 Modélisation de la dynamique future

II.3.1 Nécessité de prédire le futur

L'ensemble des accélérations atteignables par les corps du robot est limité par leurs positions actuelles ainsi que les contraintes sur le CoP. Plus la distance entre le CoP les limites du polygone de support est faible, plus les accélérations possibles sans faire basculer le robot sont limitées. Également, plus la hauteur des CoMs est grande, plus les accélérations réalisables sont limitées.

Il existe deux solutions pour permettre au robot de réaliser de plus grandes accélérations : abaisser en avance la hauteur des CoM des corps et/ou déplacer leurs positions horizontales en avance.

Autoriser au robot de réaliser de plus grandes accélérations permet d’avoir une plus grande stabilité car l’ensemble atteignable des forces de contact pour contrôler le robot, et donc la capacité à rejeter les éventuelles perturbations, est plus grand. Cela permet également au robot de réaliser des mouvements demandés plus dynamiques, et donc d’accroître son potentiel d’action.

Nous ne considérerons pas dans la suite de variation de hauteur des CoMs car nous avons, dans la section (II.2) contraint les corps à être à hauteur constante. Cependant, nous avons la possibilité de les déplacer sur le plan (\vec{x}, \vec{y}) en avance de la bonne manière.

Cette notion de “déplacer en avance de la bonne manière” implique le fait de prédire le comportement futur du robot. Afin de réaliser cela, nous avons besoin dans un premier temps de connaître les mouvements de référence demandés au robot dans le futur. Nous allons supposer, dans nos expériences, que ceux-ci sont toujours prédéfinis à l’avance.

Enfin, nous avons besoin de réaliser un modèle prédictif du robot, afin de simuler sa dynamique dans le futur.

II.3.2 Choix de la dynamique d’extrapolation

La première étape dans la réalisation d’un modèle prédictif est de définir les relations différentielles des corps d’un instant à l’autre. Afin de permettre une résolution numérique du système, et de suivre la volonté de réaliser un modèle linéaire vis à vis des variables de commande, comme réalisé en section (II.2), nous avons fait le choix d’une dynamique échantillonnée en temps, de période T , utilisant la méthode d’Euler de résolution des équations différentielles. De plus, afin d’avoir un comportement lisse du robot, et de rejeter les élasticités parasites, nous avons choisi d’utiliser une dynamique d’ordre 3, permettant une variation continue des forces de contact, et donc du CoP.

Soit v_k une variable dynamique à l’instant k , nous pouvons écrire, en utilisant la méthode d’Euler :

$$\begin{pmatrix} v_{k+T} \\ \dot{v}_{k+T} \\ \ddot{v}_{k+T} \end{pmatrix} = \begin{pmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_k \\ \dot{v}_k \\ \ddot{v}_k \end{pmatrix} + \begin{pmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{pmatrix} \ddot{v}_k \quad (\text{II.103})$$

Soit l’instant 0 étant l’instant courant, et n étant l’instant final de l’horizon de prédiction, on

note, pour toute variable dynamique v :

$$\hat{v} = \begin{pmatrix} v_0 & \dot{v}_0 & \ddot{v}_0 \end{pmatrix}^t \quad (\text{II.104})$$

et on considère ensuite, sa notation majuscule Υ défini comme suit :

$$\Upsilon = \begin{pmatrix} v_1 & \dots & v_n \end{pmatrix}^t \quad (\text{II.105})$$

$$\dot{\Upsilon} = \begin{pmatrix} \dot{v}_1 & \dots & \dot{v}_n \end{pmatrix}^t \quad (\text{II.106})$$

$$\ddot{\Upsilon} = \begin{pmatrix} \ddot{v}_1 & \dots & \ddot{v}_n \end{pmatrix}^t \quad (\text{II.107})$$

$$\ddot{\ddot{\Upsilon}} = \begin{pmatrix} \ddot{\ddot{v}}_1 & \dots & \ddot{\ddot{v}}_n \end{pmatrix}^t \quad (\text{II.108})$$

En utilisant les équations (II.103), (II.104) et (II.105)(II.106)(II.107)(II.108), nous pouvons établir les relations entre les variables Υ , $\dot{\Upsilon}$ et $\ddot{\Upsilon}$ avec les entrées $\ddot{\ddot{\Upsilon}}$ et \hat{v} de la façon suivante :

$$\Upsilon = U_v \ddot{\ddot{\Upsilon}} + S_v \hat{v} \quad (\text{II.109})$$

$$\dot{\Upsilon} = U_{\dot{v}} \ddot{\ddot{\Upsilon}} + S_{\dot{v}} \hat{v} \quad (\text{II.110})$$

$$\ddot{\Upsilon} = U_{\ddot{v}} \ddot{\ddot{\Upsilon}} + S_{\ddot{v}} \hat{v} \quad (\text{II.111})$$

avec :

$$U_v = \begin{pmatrix} \frac{T^3}{6} & 0 & 0 \\ \vdots & \ddots & 0 \\ \frac{(1-3n+3n^2)T^3}{6} & \dots & \frac{T^3}{6} \end{pmatrix}, S_v = \begin{pmatrix} 1 & T & \frac{T^2}{2} \\ \vdots & \vdots & \vdots \\ 1 & nT & \frac{n^2 T^2}{2} \end{pmatrix} \quad (\text{II.112})$$

$$U_{\dot{v}} = \begin{pmatrix} \frac{T^2}{2} & 0 & 0 \\ \vdots & \ddots & 0 \\ \frac{(2n-1)T^2}{2} & \dots & \frac{T^2}{2} \end{pmatrix}, S_{\dot{v}} = \begin{pmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & nT \end{pmatrix} \quad (\text{II.113})$$

$$U_{\ddot{v}} = \begin{pmatrix} T & 0 & 0 \\ \vdots & \ddots & 0 \\ T & \dots & T \end{pmatrix}, S_{\ddot{v}} = \begin{pmatrix} 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{II.114})$$

II.3.3 Formulation du modèle prédictif

II.3.3.1 Cas où les trois roues sont en contact avec le sol

Dans le cas où le robot possède les trois roues en contact avec le sol, on définit le jeu de variable X :

$$X = \left(\ddot{c}_1^x \quad \dots \quad \ddot{c}_n^x \quad \ddot{c}_1^y \quad \dots \quad \ddot{c}_n^y \quad \ddot{b}_1^x \quad \dots \quad \ddot{b}_n^x \quad \ddot{b}_1^y \quad \dots \quad \ddot{b}_n^y \right)^t \quad (\text{II.115})$$

On peut exprimer dans un premier temps les relations différentielles pour les variables b^{xy} :

$$B^{xy} = U_b \ddot{B}^{xy} + S_b \hat{b}^{xy} \quad (\text{II.116})$$

$$\dot{B}^{xy} = U_{\dot{b}} \ddot{B}^{xy} + S_{\dot{b}} \hat{b}^{xy} \quad (\text{II.117})$$

$$\ddot{B}^{xy} = U_{\ddot{b}} \ddot{B}^{xy} + S_{\ddot{b}} \hat{b}^{xy} \quad (\text{II.118})$$

avec B^{xy} , \dot{B}^{xy} , \ddot{B}^{xy} , \ddot{B}^{xy} et \hat{b}^{xy} définis à la manière des équations (II.105)(II.106)(II.107), (II.108) et (II.104). U_b , S_b , $U_{\dot{b}}$, $S_{\dot{b}}$, $U_{\ddot{b}}$ et $S_{\ddot{b}}$ sont quant à eux définis à la manière des équations (II.112) (II.113) (II.114).

On procède de la même manière pour les variables c^{xy} :

$$C^{xy} = U_c \ddot{C}^{xy} + S_c \hat{c}^{xy} \quad (\text{II.119})$$

$$\dot{C}^{xy} = U_{\dot{c}} \ddot{C}^{xy} + S_{\dot{c}} \hat{c}^{xy} \quad (\text{II.120})$$

$$\ddot{C}^{xy} = U_{\ddot{c}} \ddot{C}^{xy} + S_{\ddot{c}} \hat{c}^{xy} \quad (\text{II.121})$$

Enfin, en utilisant l'équation (II.78), on peut formuler la position du CoP sur l'horizon de prédiction :

$$D^{xy} = U_{dc} \ddot{C}^{xy} + S_{dc} \hat{c}^{xy} + U_{db} \ddot{B}^{xy} + S_{db} \hat{b}^{xy} + S_{dg} g^{xy} \quad (\text{II.122})$$

avec :

$$U_{db} = \frac{m_b g^z U_b + m_b h_b U_{\dot{b}}}{(m_b + m_c) g^z}, \quad S_{db} = \frac{m_b g^z S_b + m_b h_b S_{\dot{b}}}{(m_b + m_c) g^z} \quad (\text{II.123})$$

$$U_{dc} = \frac{m_c g^z U_c + m_c h_c U_{\dot{c}}}{(m_b + m_c) g^z}, \quad S_{dc} = \frac{m_c g^z S_c + m_c h_c S_{\dot{c}}}{(m_b + m_c) g^z} \quad (\text{II.124})$$

$$S_{dg} = -\frac{m_b h_b + m_c h_c}{(m_b + m_c) g^z} \quad (\text{II.125})$$

et D^{xy} défini à la manière de l'équation (II.105).

II.3.3.2 Cas où le robot bascule sur deux roues

II.3.3.2.a Relations différentielles des variables dynamiques

Dans le cas où le robot bascule sur deux roues, on a fixé le CoP à être au centre du segment de support. Nous allons formuler la dynamique de prédiction dans le repère W_k (On rappelle que l'indice $k \in \{f, r, l\}$ correspond à la direction de basculement). De plus, les équations (II.100)(II.101) montrent que les variables b^{xy} et c^{xy} sont couplées. On peut le voir plus facilement en remarquant que l'équation (II.102), qui est une combinaison linéaire des équations (II.100)(II.101), ne comporte pas les variables b^{t_k} et c^{t_k} . Cela correspond au fait que le robot est contraint à se déplacer selon l'axe \vec{n}_k , afin de garder le CoP au centre du segment de support.

Nous avons donc le jeu de variable suivant :

$$X' = \left(\ddot{b}_1^{n_k} \quad \dots \quad \ddot{b}_n^{n_k} \quad \ddot{c}_1^{n_k} \quad \dots \quad \ddot{c}_n^{n_k} \right)^t \quad (\text{II.126})$$

On peut exprimer dans un premier temps les relations différentielles pour la variable ψ_k :

$$\Psi_k = U_\psi \ddot{\Psi}_k + S_\psi \hat{\psi}_k \quad (\text{II.127})$$

$$\dot{\Psi}_k = U_{\dot{\psi}} \ddot{\Psi}_k + S_{\dot{\psi}} \hat{\psi}_k \quad (\text{II.128})$$

$$\ddot{\Psi}_k = U_{\ddot{\psi}} \ddot{\Psi}_k + S_{\ddot{\psi}} \hat{\psi}_k \quad (\text{II.129})$$

avec Ψ_k , $\dot{\Psi}_k$, $\ddot{\Psi}_k$, $\ddot{\Psi}_k$ et $\hat{\psi}_k$ définis à la manière des équations (II.105)(II.106)(II.107), (II.108) et (II.104). U_ψ , S_ψ , $U_{\dot{\psi}}$, $S_{\dot{\psi}}$, $U_{\ddot{\psi}}$ et $S_{\ddot{\psi}}$ sont quant à eux définis à la manière des équations (II.112) (II.113) (II.114).

On procède de la même manière pour la variable c^n :

$$C^m = U_c \ddot{C}^n + S_c \hat{c}^n \quad (\text{II.130})$$

$$\dot{C}^m = U_{\dot{c}} \ddot{C}^n + S_{\dot{c}} \hat{c}^n \quad (\text{II.131})$$

$$\ddot{C}^m = U_{\ddot{c}} \ddot{C}^n + S_{\ddot{c}} \hat{c}^n \quad (\text{II.132})$$

II.3.3.2.b Formulation de l'équation prédictive de la dynamique

En utilisant l'équation (II.102), on peut formuler l'équation de l'angle de basculement sur l'horizon de prédiction :

$$\begin{aligned}
& - \left((m_b h_b + m_c h_c) g^z U_\psi + (m_b h_b^2 + m_c h_c^2) U_{\ddot{\psi}} \right) \ddot{\Psi} \\
& = (m_b h_b U_{\ddot{b}} + m_c g^z U_b) \ddot{\ddot{B}}^n + (m_c h_c U_{\ddot{c}} + m_c g U_c) \ddot{\ddot{C}}^n \\
& + (m_b h_b S_{\ddot{b}} + m_c g^z S_b) \hat{\delta}^n + (m_c h_c S_{\ddot{c}} + m_c g^z S_c) \hat{c}^n \\
& - (m_b h_b + m_c h_c) \left(\cos(\theta_k) g^x + \sin(\theta_k) g^y \right) \\
& + \left((m_b h_b + m_c h_c) g^z S_\psi + (m_b h_b^2 + m_c h_c^2) S_{\ddot{\psi}} \right) \hat{\psi}
\end{aligned} \tag{II.133}$$

Afin de clarifier l'équation (II.133), on la réécrit de la façon suivante :

$$\tilde{U}_\psi \ddot{\Psi} = U_{\psi b} \ddot{\ddot{B}}^n + U_{\psi c} \ddot{\ddot{C}}^n + S_{\psi b} \hat{\delta}^n + S_{\psi c} \hat{c}^n + S_{\psi g^x} g^x + S_{\psi g^y} g^y + S_{\psi \psi} \hat{\psi} \tag{II.134}$$

avec :

$$\tilde{U}_\psi = -(m_b h_b + m_c h_c) g^z U_\psi - (m_b h_b^2 + m_c h_c^2) U_{\ddot{\psi}} \tag{II.135}$$

$$U_{\psi b} = m_b h_b U_{\ddot{b}} + m_c g^z U_b \tag{II.136}$$

$$U_{\psi c} = m_c h_c U_{\ddot{c}} + m_c g U_c \tag{II.137}$$

$$S_{\psi b} = m_b h_b S_{\ddot{b}} + m_c g^z S_b \tag{II.138}$$

$$S_{\psi c} = m_c h_c S_{\ddot{c}} + m_c g^z S_c \tag{II.139}$$

$$S_{\psi g^x} = -(m_b h_b + m_c h_c) \cos(\theta_k) \tag{II.140}$$

$$S_{\psi g^y} = -(m_b h_b + m_c h_c) \sin(\theta_k) \tag{II.141}$$

$$S_{\psi \psi} = (m_b h_b + m_c h_c) g^z S_\psi + (m_b h_b^2 + m_c h_c^2) S_{\ddot{\psi}} \tag{II.142}$$

II.3.3.2.c Inversibilité de la dynamique de basculement

On peut enfin remarquer que la dérivée troisième de l'angle de basculement $\ddot{\Psi}$ sur l'horizon de prédiction se formule de manière directe avec les variables de commande $\ddot{\ddot{B}}^n$ et $\ddot{\ddot{C}}^n$ si et seulement si \tilde{U}_ψ est inversible.

On peut noter que les matrices U_ψ et $U_{\ddot{\psi}}$ sont triangulaires inférieures avec une diagonale constante de valeurs respectives $\frac{T^3}{6}$ et T . Ainsi, \tilde{U}_ψ étant une somme pondérée par des scalaires de ces deux matrices, elle est inversible si et seulement si :

$$-(m_b h_b + m_c h_c)g^z \frac{T^3}{6} - (m_b h_b^2 + m_c h_c^2)T \neq 0 \quad (\text{II.143})$$

En résolvant l'équation (II.143), on peut déterminer que \tilde{U}_ψ est inversible si les conditions suivantes sont respectées :

$$T \neq \sqrt{\frac{m_b h_b^2 + m_c h_c^2}{-6(m_b h_b + m_c h_c)g^z}} \quad (\text{II.144})$$

$$T > 0 \quad (\text{II.145})$$

$$-(m_b h_b + m_c h_c)g^z > 0 \quad (\text{II.146})$$

Notamment, on peut remarquer que la dynamique de basculement n'est plus inversible lorsque le robot se retrouve en position horizontale, car l'équation (II.146) ne serait plus respectée. De plus, pour des raisons de stabilité numérique, il ne faut pas choisir une période d'échantillonnage T proche de la valeur interdite de l'équation (II.144). En utilisant les paramètres du robot présenté en annexe (A), nous obtenons $T \neq 100,4$ ms.

II.3.4 Synthèse

Dans cette section, nous avons défini un modèle de prédiction du futur de la dynamique de notre système. Nous nous sommes basé sur une dynamique échantillonnée d'ordre 3, utilisant la méthode d'Euler de résolution des équations différentielles. Ensuite de quoi, nous avons pu itérer cette dynamique sur un nombre d'échantillons n , et l'appliquer aux deux modèles du système : lorsque le robot est sur trois roues (II.78), et lorsqu'il bascule sur deux (II.102).

Les équations résultantes (II.122)(II.134) seront utilisées dans le chapitre III afin de commander le système.

Chapitre III

Commande par modèle prédictif

III.1 Principe

L'objectif de ce chapitre est de présenter la loi de commande permettant de réaliser les mouvements et l'équilibre d'un robot humanoïde à roues omnidirectionnelles. Dans le chapitre II, nous avons développé un modèle du robot utilisant deux corps. Les contraintes de complémentarités dues aux forces de contact nous ont amené à modéliser le robot en deux parties : lorsque celui-ci possède ses trois roues en contact avec le sol, et lorsqu'il bascule sur deux de ses roues. Ensuite de quoi, nous avons montré qu'il est important de modéliser la dynamique dans le futur à cause des contraintes sur le CoP, qui limitent fortement les mouvements réalisables par le robot. Enfin, deux équations prédictives de la dynamique ont été formulées :

- Dans le cas où le robot possède trois roues sur le sol, l'équation (II.122) exprime la relation entre la position du CoP et la dynamique des corps du robot.
- Dans le cas où le robot bascule sur deux roues, l'équation (II.134) exprime la dynamique de l'angle de basculement en fonction de celle des corps du robot.

Nous avons donc choisi de réaliser une loi de commande optimale basée sur un modèle prédictif linéaire vis à vis des variables de commande, soumis à des contraintes linéaires. La fonction objectif de cette commande optimale est basée sur la minimisation d'une norme d'ordre 2, permettant la convergence d'une grandeur d'erreur vers 0, utilisés notamment dans une tâche de suivi de trajectoire.

L'intérêt de l'aspect prédictif de cette commande est de pouvoir utiliser les connaissances que l'on a sur le comportement dynamique du système ainsi que sur les mouvements demandés, pour prévoir et adapter en avance les trajectoires résultantes. Cela permet d'obtenir une meilleure réalisation des objectifs dans le temps. Cette adaptation est nécessaire lorsque les contraintes du système réduisent de beaucoup l'ensemble des trajectoires atteignables instantanément par les corps du robot, ce qui est notre cas.

III.2 Outil mathématique et contraintes associées

Notre objectif est d'exécuter la loi de commande en temps réel sur la plate-forme expérimentale. Il est donc important d'utiliser un outil mathématique de calcul qui permette cela. Le système étant fortement contraint (Position du CoP, limites articulaires, limites dynamique des moteurs), nous ne pouvons disposer d'une solution analytique. Une méthode de résolution numérique du problème doit donc être envisagée.

Nous avons donc choisi d'utiliser la méthode de la programmation quadratique sous contraintes linéaires (QP). Cette méthode de formulation d'un problème d'optimisation permet de minimiser une fonction objectif quadratique vis à vis des variables d'optimisation, celles-ci étant soumises à des contraintes d'inégalités linéaires :

$$\begin{cases} \min_x (x^t Q x + \rho^t x) \\ v^- \leq V x \leq v^+ \end{cases} \quad (\text{III.1})$$

avec x le vecteur des variables à optimiser, Q la matrice hessienne, ρ le vecteur linéaire, V la matrice des contraintes et v^+ et v^- les vecteurs des bornes des contraintes.

Une fois le problème posé, un solveur de QP permet, en temps réel, de calculer la solution du problème de minimisation en un certain nombre d'itérations, dépendant principalement des contraintes.

Il existe différentes méthodes afin de résoudre un problème quadratique [9]. Le choix d'une méthode dépend fortement des caractéristiques du problème à réaliser (nombre de variables et de contraintes, structure de la matrice hessienne, structure des contraintes, etc). Ce choix va impacter le temps de calcul nécessaire afin de résoudre le problème.

Dans notre cas, le système à résoudre varie peu d'un instant de calcul à l'autre. Il est donc possible d'utiliser des stratégies de warm-start, consistant à donner une estimation de la solution optimale, afin de réduire le nombre d'itérations nécessaires pour résoudre le problème. Cette stratégie de warm-start est basée sur la solution calculée au pas de temps précédent.

En pratique, le nombre d'itérations est réduit à un ou deux, en l'absence de perturbations agissant sur le robot. De ce fait, le choix de la méthode de résolution n'impacte que peu le temps de calcul, car le problème d'optimisation est résolu la grande majorité du temps en une ou deux itérations. Le choix s'est porté sur le solveur qpOASES [12] pour des questions de disponibilités vis à vis de l'entreprise. Pour information, un autre solveur de QP a été testé : LSSOL [35]. L'implémentation de LSSOL est environs 20% plus performante en terme de temps de calcul que celle de qpOASES.

III.3 Formulation des problèmes d'optimisations

III.3.1 Introduction

De par les contraintes de complémentarité mixtes du modèle (II.59), ainsi que des contraintes cinématiques du robot et de la base mobile, il n'est pas possible de résoudre simplement l'équation de la dynamique. Nous avons donc choisi de présenter la commande du robot sous la forme de trois contrôleurs supervisés par une machine à état.

Le premier contrôleur correspond au cas où le robot possède les trois roues sur le sol, en utilisant la dynamique de l'équation (II.122). En l'absence de perturbation emmenant le CoP sur le bord du polygone de support, ce contrôleur seul est suffisant et assure que le robot ne basculera pas lors de ses mouvements.

Le problème se corse lorsqu'une perturbation fait basculer le robot. Nous avons donc choisi de définir deux autres contrôleurs permettant de gérer cette situation. Le second correspond au moment où le robot est en train de basculer, en utilisant la dynamique de l'équation (II.134). Le dernier permet d'assurer une transition cohérente entre les deux modèles.

Le modèle dynamique de basculement (II.134) ne prend pas en compte la présence du sol, qui ajoute une force compensant la gravité lorsque l'angle de basculement atteint 0. Cette absence d'information emmène le second contrôleur à se comporter de manière non désirée à de faibles

angles de basculement, en compensant la gravité par l'accélération de la base mobile, au lieu de "laisser tomber" le robot sur le sol en laissant les forces de contact la compenser par la suite. Ce troisième contrôleur est là pour gérer ce cas particulier afin d'assurer une bonne transition.

III.3.2 Lorsque les trois roues sont en contact avec le sol

III.3.2.1 Formulation des objectifs

Lorsque les trois roues sont en contact avec le sol, nous pouvons définir trois objectifs de commande :

- Assurer le suivi d'une trajectoire de référence. Nous avons fait le choix d'un asservissement position/vitesse de la trajectoire de la base mobile.
- Assurer une bonne robustesse aux perturbations. Cet objectif consiste à accroître le potentiel d'action du robot. Ce potentiel d'action est limité par les contraintes sur le CoP D^{xy} , ainsi que les contraintes cinématiques entre le corps du robot et la base mobile. En optimisant au plus loin possible le CoP D^{xy} ainsi que le CoM C^{xy} du corps du robot de leurs contraintes respectives, nous assurons une robustesse optimale face à une perturbation inconnue.
- Assurer la stabilité numérique de la commande optimale. Cela peut être réalisé en minimisant le carré des variables de commande X .

Dans la suite, nous noterons O_i l'objectif i . Il correspond à la norme élevée au carré d'une fonction linéaire des variables de commande. Il s'écrit de la façon suivante :

$$O_i = \frac{1}{2} \left\| f_i(X) \right\|^2 \quad (\text{III.2})$$

puis, en utilisant la notation des problèmes quadratiques :

$$O_i = \frac{1}{2} X^t Q_i X + \rho_i^t X + \epsilon_i \quad (\text{III.3})$$

avec Q_i une matrice symétrique définie positive (dite Hessienne), ρ_i un vecteur et ϵ_i un scalaire constant. Nous ne détaillerons pas l'expression de ϵ_i car ce terme n'a pas d'impact pas dans la minimisation des l'objectif O_i (Plus de détails en section (III.3.2.3)).

On rappelle le vecteur de commande dans le cas où le robot à les trois roues en contact avec le sol :

$$X = \begin{pmatrix} \ddot{C}^x & \ddot{C}^y & \ddot{B}^x & \ddot{B}^y \end{pmatrix}^t \quad (\text{III.4})$$

III.3.2.1.a Suivi de trajectoire

L'objectif principal de la commande est d'assurer un suivi de trajectoire de la base mobile, par rapport à une trajectoire de référence $(B_{ref}^{xy}, \dot{B}_{ref}^{xy})$ définie sur l'ensemble de l'horizon de prédiction.

Nous pouvons donc écrire les objectifs O_1 et O_2 , qui correspondent respectivement au suivi de trajectoire en position et en vitesse :

$$O_1 = \frac{1}{2} \left\| B^{xy} - B_{ref}^{xy} \right\|^2 = \frac{1}{2} X^t Q_1 X + \rho_1^t X + \epsilon_1 \quad (\text{III.5})$$

$$O_2 = \frac{1}{2} \left\| \dot{B}^{xy} - \dot{B}_{ref}^{xy} \right\|^2 = \frac{1}{2} X^t Q_2 X + \rho_2^t X + \epsilon_2 \quad (\text{III.6})$$

avec :

$$Q_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & U_b^t U_b & 0 \\ 0 & 0 & 0 & U_b^t U_b \end{pmatrix}, \quad \rho_1 = \begin{pmatrix} 0 \\ 0 \\ U_b^t (S_b \hat{b}^x - B_{ref}^x) \\ U_b^t (S_b \hat{b}^y - B_{ref}^y) \end{pmatrix} \quad (\text{III.7})$$

$$Q_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & U_b^t U_b & 0 \\ 0 & 0 & 0 & U_b^t U_b \end{pmatrix}, \quad \rho_2 = \begin{pmatrix} 0 \\ 0 \\ U_b^t (S_b \hat{b}^x - \dot{B}_{ref}^x) \\ U_b^t (S_b \hat{b}^y - \dot{B}_{ref}^y) \end{pmatrix} \quad (\text{III.8})$$

III.3.2.1.b Robustesse aux perturbations

L'objectif suivant de la commande est d'assurer une bonne robustesse face à une perturbation inconnue. Dans ce cas, la meilleure robustesse est atteinte lorsque le CoP D^{xy} et le CoM du corps du robot C^{xy} est au plus loin des contraintes, donc au centre de la base mobile aux points B^{xy} .

Nous pouvons donc définir deux objectifs O_3 et O_4 , qui correspondent respectivement au cen-

trage du CoP et à celui du CoM du corps du robot :

$$O_3 = \frac{1}{2} \left\| D^{xy} - B^{xy} \right\|^2 = \frac{1}{2} X^t Q_3 X + \rho_3^t X + \epsilon_3 \quad (\text{III.9})$$

$$O_4 = \frac{1}{2} \left\| C^{xy} - B^{xy} \right\|^2 = \frac{1}{2} X^t Q_4 X + \rho_4^t X + \epsilon_4 \quad (\text{III.10})$$

avec :

$$Q_3 = \begin{pmatrix} U_{dc}^t U_{dc} & 0 & U_{dc}^t (U_{db} - U_b) & 0 \\ 0 & U_{dc}^t U_{dc} & 0 & U_{dc}^t (U_{db} - U_b) \\ (U_{db}^t - U_b^t) U_{dc} & 0 & (U_{db}^t - U_b^t) (U_{db} - U_b) & 0 \\ 0 & (U_{db}^t - U_b^t) U_{dc} & 0 & (U_{db}^t - U_b^t) (U_{db} - U_b) \end{pmatrix}$$

$$\rho_3 = \begin{pmatrix} U_{dc}^t \left(S_{dc} \hat{c}^x + (S_{db} - S_b) \hat{b}^x + S_{dg} g^x \right) \\ U_{dc}^t \left(S_{dc} \hat{c}^y + (S_{db} - S_b) \hat{b}^y + S_{dg} g^y \right) \\ \left(U_{db}^t - U_b^t \right) \left(S_{dc} \hat{c}^x + (S_{db} - S_b) \hat{b}^x + S_{dg} g^x \right) \\ \left(U_{db}^t - U_b^t \right) \left(S_{dc} \hat{c}^y + (S_{db} - S_b) \hat{b}^y + S_{dg} g^y \right) \end{pmatrix} \quad (\text{III.11})$$

$$Q_4 = \begin{pmatrix} U_c^t U_c & 0 & -U_c^t U_b & 0 \\ 0 & U_c^t U_c & 0 & -U_c^t U_b \\ -U_b^t U_c & 0 & U_b^t U_b & 0 \\ 0 & -U_b^t U_c & 0 & U_b^t U_b \end{pmatrix}, \quad \rho_4 = \begin{pmatrix} U_c^t S_b \hat{b}^x \\ U_c^t S_b \hat{b}^y \\ U_b^t S_b \hat{b}^x \\ U_b^t S_b \hat{b}^y \end{pmatrix} \quad (\text{III.12})$$

III.3.2.1.c Stabilité numérique

Le dernier objectif concerne la stabilité numérique du problème d'optimisation. Afin de réaliser cela, une solution consiste à minimiser le jerk des commandes \ddot{C}^{xy} et \ddot{B}^{xy} .

Nous pouvons donc écrire le dernier objectif O_5 :

$$O_5 = \frac{1}{2} \left\| X \right\|^2 = \frac{1}{2} X^t Q_5 X + \rho_5^t X + \epsilon_5 \quad (\text{III.13})$$

avec :

$$Q_5 = \mathcal{I}, \quad \rho_5 = 0 \quad (\text{III.14})$$

où \mathcal{I} représente une matrice identité.

III.3.2.2 Formulation des contraintes

Lorsque les trois roues sont en contact avec le sol, il est important de définir un certain nombre de contraintes :

- Assurer l'équilibre du robot, et donc ne pas le mettre en situation de basculement. Cette contrainte s'exprime via le CoP D^{xy} .
- Respecter les limites physiques de la base mobile en limitant les vitesses et accélérations maximales.
- Respecter la cinématique du corps du robot, en limitant la distance entre son CoM C^{xy} et celui de la base mobile B^{xy} .

La commande ne prend pas en compte la rotation du robot autour de l'axe \vec{z} , qui modifie la forme des contraintes. Nous allons donc systématiquement dans la suite réduire celle-ci de façon conservative à un cercle, qui est invariant par rotation autour de son centre.

Enfin, une contrainte circulaire étant non-linéaire, elle ne peut pas s'exprimer directement à l'aide d'inégalités linéaires. La dernière étape afin de formaliser la contrainte va être de la réduire à un polygone, de façon conservative également. L'ordre de ce polygone va impacter sur le temps de calcul de la commande : Plus celui-ci est grand, plus on est proche d'une contrainte invariante par rotation, mais plus le temps de calcul est grand. Le choix de cet ordre va être déterminé par la criticité de chaque contrainte vis à vis de l'équilibre et du comportement général du robot.

Dans les sections suivantes, nous écrirons les contraintes linéarisées de la façon suivante :

$$v_i^- \leq V_i X \leq v_i^+ \quad (\text{III.15})$$

avec v_i^- et v_i^+ les vecteurs bornes de la contrainte i et V la matrice de la contrainte i .

III.3.2.2.a Contraintes de non-basculement

La contrainte principale à la stabilité de la commande est celle empêchant le robot de basculer [Fig. III.1]. Nous rappelons l'expression des contraintes de non-basculement (II.79)(II.80)(II.81).

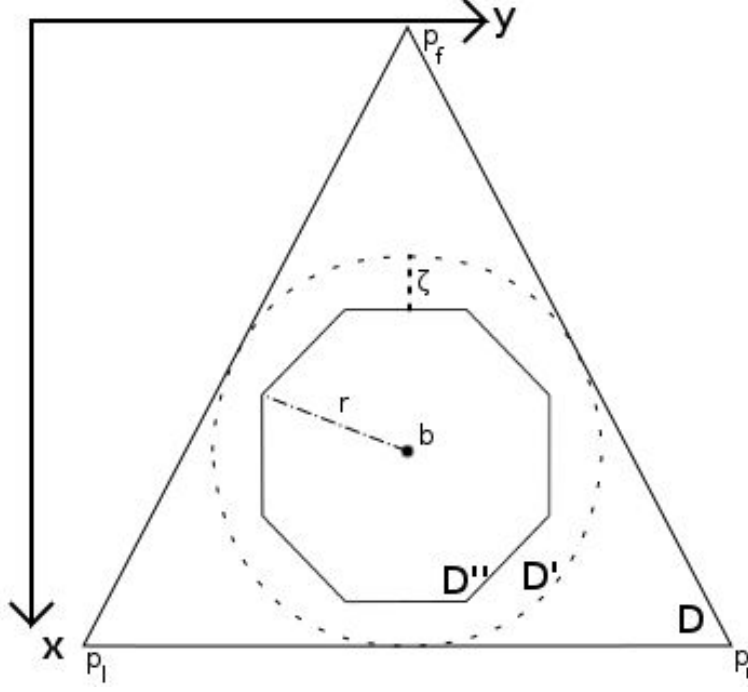


FIGURE III.1 – Représentation des contraintes de non-basculement. D correspond à la contrainte exacte. D' correspond à la contrainte circulaire inscrite dans D , rendant celle-ci invariante par rotation autour de b . D'' correspond à la linéarisation de la contrainte D' par un octogone, en ajoutant une marge de sécurité ζ .

$\forall k \in (1, n)$:

$$d_k^{xy} \times (p_r^{xy} - p_f^{xy}) > 0 \quad (\text{III.16})$$

$$d_k^{xy} \times (p_l^{xy} - p_r^{xy}) > 0 \quad (\text{III.17})$$

$$d_k^{xy} \times (p_f^{xy} - p_l^{xy}) > 0 \quad (\text{III.18})$$

Ces contraintes ne sont pas invariantes par rotation de la base mobile. On réduit donc celles-ci à un cercle d'origine B^{xy} que l'on linéarise par la suite par un octogone régulier de rayon r' , orienté

selon la direction \vec{x} [Fig. III.1] :

$$r' = \min(d_f, d_r, d_l) - \zeta \quad (\text{III.19})$$

$$\zeta > 0 \quad (\text{III.20})$$

où ζ est une marge de sécurité. La formulation QP ne permet pas l'utilisation de contraintes d'in-égalités strictes, la marge de sécurité ζ permet de s'en affranchir.

Les contraintes linéarisées s'expriment donc directement de la façon suivante :

$$v_1^- \leq V_1 X \leq v_1^+ \quad (\text{III.21})$$

avec

$$\begin{aligned} V_1 &= \begin{pmatrix} U_{dc} & 0 & U_{db} - U_b & 0 \\ 0 & U_{dc} & 0 & U_{db} - U_b \\ U_{dc} & U_{dc} & U_{db} - U_b & U_{db} - U_b \\ U_{dc} & -U_{dc} & U_{db} - U_b & -U_{db} + U_b \end{pmatrix} \\ v_1^- &= \begin{pmatrix} -3\tilde{r} - S_{dc}\hat{c}^x - (S_{db} - U_b)\hat{b}^x \\ -3\tilde{r} - S_{dc}\hat{c}^y - (S_{db} - U_b)\hat{b}^y \\ -4\tilde{r} - S_{dc}(\hat{c}^x + \hat{c}^y) - (S_{db} - U_b)(\hat{b}^x + \hat{b}^y) \\ -4\tilde{r} - S_{dc}(\hat{c}^x - \hat{c}^y) - (S_{db} - U_b)(\hat{b}^x - \hat{b}^y) \end{pmatrix} \\ v_1^+ &= \begin{pmatrix} 3\tilde{r} - S_{dc}\hat{c}^x - (S_{db} - U_b)\hat{b}^x \\ 3\tilde{r} - S_{dc}\hat{c}^y - (S_{db} - U_b)\hat{b}^y \\ 4\tilde{r} - S_{dc}(\hat{c}^x + \hat{c}^y) - (S_{db} - U_b)(\hat{b}^x + \hat{b}^y) \\ 4\tilde{r} - S_{dc}(\hat{c}^x - \hat{c}^y) - (S_{db} - U_b)(\hat{b}^x - \hat{b}^y) \end{pmatrix} \end{aligned} \quad (\text{III.22})$$

$$\text{et } \tilde{r} = \frac{r'}{\sqrt{10}}.$$

III.3.2.2.b Vitesses et accélérations maximales de la base mobile

Les moteurs des roues de la base mobile sont limités en vitesse et accélération. Il faut donc prendre en compte cet élément afin de ne pas générer de commandes infaisables par le robot. Il n'y a pas de relation linéaire entre les limites cinématiques des moteurs des roues et celles de la base

mobile. Nous allons donc définir des contraintes concernant la base mobile, conservatives vis à vis de celles des moteurs.

Soit \dot{b}_m et \ddot{b}_m des limites en vitesse et accélération de la base mobile, nous pouvons exprimer les contraintes de la façon suivante. $\forall k \in (1, n)$:

$$\left\| \vec{\dot{b}}_k \right\| \leq \dot{b}_m \quad (\text{III.23})$$

$$\left\| \vec{\ddot{b}}_k \right\| \leq \ddot{b}_m \quad (\text{III.24})$$

Ces contraintes étant non-linéaires, nous avons choisi de les approximer simplement par un carré, car, contrairement aux contraintes sur le CoP, celles-ci ne sont pas critiques concernant la stabilité du robot. Ainsi, même si les contraintes linéaires résultantes sont significativement non-conservatives par rotation du robot, cela n'impacte que de façon mineure le comportement de la commande. Cette solution est donc préférable, du point de vue du temps de calcul de la commande, au choix d'un polygone d'ordre plus élevé.

Les contraintes linéarisées s'expriment donc de la façon suivante :

$$v_2^- \leq V_2 X \leq v_2^+ \quad (\text{III.25})$$

avec :

$$V_2 = \begin{pmatrix} 0 & 0 & U_{\dot{b}} & 0 \\ 0 & 0 & 0 & U_{\dot{b}} \\ 0 & 0 & U_{\ddot{b}} & 0 \\ 0 & 0 & 0 & U_{\ddot{b}} \end{pmatrix}, \quad v_2^- = \begin{pmatrix} -\tilde{\dot{b}}_m - S_{\dot{b}} \hat{b}^x \\ -\tilde{\dot{b}}_m - S_{\dot{b}} \hat{b}^y \\ -\tilde{\ddot{b}}_m - S_{\ddot{b}} \hat{b}^x \\ -\tilde{\ddot{b}}_m - S_{\ddot{b}} \hat{b}^y \end{pmatrix}, \quad v_2^+ = \begin{pmatrix} \tilde{\dot{b}}_m - S_{\dot{b}} \hat{b}^x \\ \tilde{\dot{b}}_m - S_{\dot{b}} \hat{b}^y \\ \tilde{\ddot{b}}_m - S_{\ddot{b}} \hat{b}^x \\ \tilde{\ddot{b}}_m - S_{\ddot{b}} \hat{b}^y \end{pmatrix} \quad (\text{III.26})$$

et :

$$\tilde{\dot{b}}_m = \frac{\sqrt{2}}{2} \dot{b}_m \quad (\text{III.27})$$

$$\tilde{\ddot{b}}_m = \frac{\sqrt{2}}{2} \ddot{b}_m \quad (\text{III.28})$$

III.3.2.2.c Limites articulaires du corps du robot

L'agencement des corps et articulations du robot implique que la distance entre son CoM C^{xy} et celui de la base mobile B^{xy} est limitée. Sur la plate-forme expérimentale, si l'on considère les approximations réalisées dans le chapitre II (c^z constant et on ignore les mouvements des bras), cette

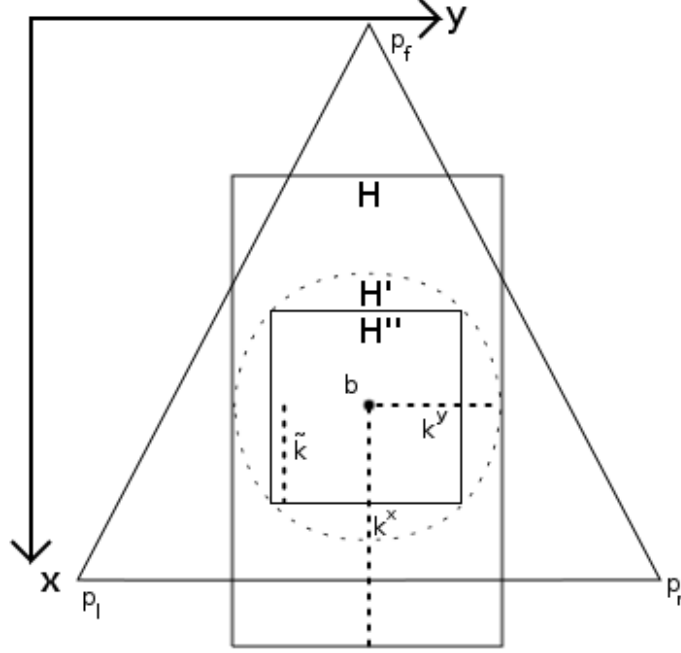


FIGURE III.2 – Représentation des contraintes cinématiques du corps du robot. H correspond à la contrainte exacte. H' correspond à la contrainte circulaire inscrite dans H , rendant celle-ci invariante par rotation autour de b . H'' correspond à la linéarisation de la contrainte H' par un carré.

zone de déplacement correspond exactement à un rectangle centré autour de B^{xy} , de dimension $(2k^x, 2k^y)$ [Fig. III.2].

Cette contrainte n'est pas invariante par rotation. Nous choisissons donc de réduire cette contrainte à un cercle centré autour de la base mobile B^{xy} , de rayon $\min(k^x, k^y)$. Enfin, on linéarise ce cercle en un carré de dimension $2\tilde{k}$ [Fig. III.2], avec :

$$\tilde{k} = \frac{\sqrt{2}}{2} \min(k^x, k^y) \quad (\text{III.29})$$

Nous ne choisissons pas ici un polygone d'ordre supérieur à 2 car les limites articulaires du corps du robot sont dans la majorité des cas négligeables devant celles concernant le CoP. Ainsi, dans une situation normale, le robot ne se retrouve généralement pas en contrainte sur ses limites

articulaires, ce qui rend un polygone d'ordre 2 suffisant pour représenter les contraintes associées.

Les contraintes linéarisées s'expriment donc de la façon suivante :

$$v_3^- \leq V_3 X \leq v_3^+ \quad (\text{III.30})$$

avec

$$V_3 = \begin{pmatrix} U_c & 0 & -U_b & 0 \\ 0 & U_c & 0 & -U_b \end{pmatrix}, \quad v_3^- = \begin{pmatrix} -\tilde{k} - S_c \hat{c}^x + S_b \hat{b}^x \\ -\tilde{k} - S_c \hat{c}^y + S_b \hat{b}^y \end{pmatrix}, \quad v_3^+ = \begin{pmatrix} \tilde{k} - S_c \hat{c}^x + S_b \hat{b}^x \\ \tilde{k} - S_c \hat{c}^y + S_b \hat{b}^y \end{pmatrix} \quad (\text{III.31})$$

III.3.2.3 Problème quadratique résultant

La loi de commande optimale présentée possède plusieurs objectifs. Il existe classiquement deux manières de résoudre un problème multi-objectifs :

- Considérer une hiérarchie d'objectifs. A chaque niveau hiérarchique, un seul objectif est optimisé, ceux qui sont prioritaires sont considérés comme des contraintes, et les autres ne sont pas optimisés à ce niveau de hiérarchie.
- Considérer une somme pondérée d'objectifs. Plus les pondérations associées sont relativement grandes les unes par rapport aux autres, plus les objectifs associés sont prioritaires.

Nous avons fait le choix de résoudre le problème par une somme pondérée d'objectifs [37]. Ce choix est motivé d'une part par le fait que les solveurs de QP que nous avons de disponibles ne permettent pas de résoudre un problème hiérarchique. D'autre part, en utilisant le principe d'une somme pondérée d'objectifs, nous pouvons tendre vers la première solution en augmentant l'écart relatif entre les pondérations de plusieurs ordres de grandeurs. Réaliser l'inverse n'est pas possible.

Soit α_i la pondération associée à l'objectif i , nous pouvons écrire le problème quadratique résultant de la façon suivante :

$$\left\{ \begin{array}{l} \min_X \left(X^t \sum_{i=1}^5 (\alpha_i Q_i) X + \sum_{i=1}^5 (\alpha_i \rho_i^t) X \right) \\ \begin{pmatrix} v_1^- \\ v_2^- \\ v_3^- \end{pmatrix} \leq \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} X \leq \begin{pmatrix} v_1^+ \\ v_2^+ \\ v_3^+ \end{pmatrix} \end{array} \right. \quad (\text{III.32})$$

III.3.3 Lorsque le robot bascule sur deux roues

III.3.3.1 Formulation des objectifs

Lorsque le robot bascule sur deux roues, nous pouvons définir trois objectifs de commande :

- Contrôler l'angle de basculement à 0, afin de ramener le robot sur trois roues.
- Minimiser la vitesse angulaire de basculement. Cet objectif a pour but de limiter l'impact occasionné lorsque l'angle de basculement atteint 0. Si la vitesse angulaire est nulle au moment où la roue touche le sol, il n'y a pas d'impact.
- Assurer la stabilité numérique de la commande optimale. Cela peut être réalisé en minimisant le carré des variables de commande X' .

Dans la suite, nous noterons O'_i l'objectif i . Il correspond à la norme élevée au carré d'une fonction linéaire des variables de commande. Il s'écrit de la façon suivante :

$$O'_i = \frac{1}{2} \left\| f'_i(X) \right\|^2 \quad (\text{III.33})$$

puis, en utilisant la notation QP :

$$O'_i = \frac{1}{2} X^t Q'_i X + \rho'^t_i X + \epsilon'_i \quad (\text{III.34})$$

avec Q'_i une matrice symétrique définie positive (dite Hessienne), ρ'_i un vecteur et ϵ_i un scalaire constant. Nous ne détaillerons pas l'expression des ϵ'_i car ce terme n'a pas d'impact pas dans la minimisation des l'objectifs O'_i .

On rappelle le vecteur de commande dans le cas où le robot bascule sur deux roues :

$$X' = \begin{pmatrix} \ddot{C}^n & \ddot{B}^n \end{pmatrix}^t \quad (\text{III.35})$$

III.3.3.1.a Minimisation de l'angle de basculement

L'objectif principal de la commande est d'assurer le retour sur trois roues au sol du robot en amenant l'angle de basculement à 0. Nous pouvons donc écrire l'objectif O'_1 correspondant en utilisant l'équation (II.134) :

$$O'_1 = \frac{1}{2} \left\| \Psi \right\|^2 = \frac{1}{2} X'^t Q'_1 X' + \rho'^t_1 X' + \epsilon'_1 \quad (\text{III.36})$$

avec :

$$\begin{aligned} Q'_1 &= \begin{pmatrix} U_{\psi c}^t \tilde{U}_{\psi}^{-t} \tilde{U}_{\psi}^{-1} U_{\psi c} & U_{\psi c}^t \tilde{U}_{\psi}^{-t} \tilde{U}_{\psi}^{-1} U_{\psi b} \\ U_{\psi b}^t \tilde{U}_{\psi}^{-t} \tilde{U}_{\psi}^{-1} U_{\psi c} & U_{\psi b}^t \tilde{U}_{\psi}^{-t} \tilde{U}_{\psi}^{-1} U_{\psi b} \end{pmatrix} \\ \rho'_1 &= \begin{pmatrix} U_{\psi c}^t \tilde{U}_{\psi}^{-t} (S_{\psi b} \hat{b}^n + S_{\psi c} \hat{c}^n + S_{\psi g^x} g^x + S_{\psi g^y} g^y + S_{\psi \psi} \hat{\psi}) \\ U_{\psi b}^t \tilde{U}_{\psi}^{-t} (S_{\psi b} \hat{b}^n + S_{\psi c} \hat{c}^n + S_{\psi g^x} g^x + S_{\psi g^y} g^y + S_{\psi \psi} \hat{\psi}) \end{pmatrix} \end{aligned} \quad (\text{III.37})$$

III.3.3.1.b Minimisation de la vitesse angulaire

Afin de limiter l'impact du robot lorsqu'il touchera le sol, il est important de minimiser la vitesse angulaire de basculement. Pour exprimer l'objectif correspondant, il faut dans un premier temps écrire les relations entre la vitesse angulaire de basculement $\dot{\Psi}$ et les variables du problème X' .

$$\dot{\Psi} = U_{\psi} \ddot{\Psi} + S_{\psi} \hat{\psi} \quad (\text{III.38})$$

En utilisant l'équation (II.134), on peut donc écrire :

$$\dot{\Psi} = U_{\psi} \tilde{U}_{\psi}^{-1} \left(U_{\psi b} \ddot{B}^n + U_{\psi c} \ddot{C}^n + S_{\psi b} \hat{b}^n + S_{\psi c} \hat{c}^n + S_{\psi g^x} g^x + S_{\psi g^y} g^y \right) + \left(U_{\psi} \tilde{U}_{\psi}^{-1} S_{\psi \psi} + S_{\psi} \right) \hat{\psi} \quad (\text{III.39})$$

Nous pouvons donc écrire l'objectif O'_2 :

$$O'_2 = \frac{1}{2} \left\| \dot{\Psi} \right\|^2 = \frac{1}{2} X'^t Q'_2 X' + \rho'^t_2 X + \epsilon'_2 \quad (\text{III.40})$$

avec :

$$\begin{aligned} Q'_2 &= \begin{pmatrix} U_{\psi c}^t \tilde{U}_{\psi}^{-t} U_{\psi}^t U_{\psi} \tilde{U}_{\psi}^{-1} U_{\psi c} & U_{\psi c}^t \tilde{U}_{\psi}^{-t} U_{\psi}^t U_{\psi} \tilde{U}_{\psi}^{-1} U_{\psi b} \\ U_{\psi b}^t \tilde{U}_{\psi}^{-t} U_{\psi}^t U_{\psi} \tilde{U}_{\psi}^{-1} U_{\psi c} & U_{\psi b}^t \tilde{U}_{\psi}^{-t} U_{\psi}^t U_{\psi} \tilde{U}_{\psi}^{-1} U_{\psi b} \end{pmatrix} \\ \rho'_2 &= \begin{pmatrix} U_{\psi c}^t \tilde{U}_{\psi}^{-t} U_{\psi}^t \left(U_{\psi} \tilde{U}_{\psi}^{-1} (S_{\psi b} \hat{b}^n + S_{\psi c} \hat{c}^n + S_{\psi g^x} g^x + S_{\psi g^y} g^y) + (U_{\psi} \tilde{U}_{\psi}^{-1} S_{\psi \psi} + S_{\psi}) \hat{\psi} \right) \\ U_{\psi b}^t \tilde{U}_{\psi}^{-t} U_{\psi}^t \left(U_{\psi} \tilde{U}_{\psi}^{-1} (S_{\psi b} \hat{b}^n + S_{\psi c} \hat{c}^n + S_{\psi g^x} g^x + S_{\psi g^y} g^y) + (U_{\psi} \tilde{U}_{\psi}^{-1} S_{\psi \psi} + S_{\psi}) \hat{\psi} \right) \end{pmatrix} \end{aligned} \quad (\text{III.41})$$

III.3.3.1.c Stabilité numérique

Le dernier objectif concerne la stabilité numérique du problème d'optimisation. Afin de réaliser cela, une solution consiste à minimiser le jerk des commandes \ddot{C}^n et \ddot{B}^n .

Nous pouvons donc écrire le dernier objectif O'_5 :

$$O'_3 = \frac{1}{2} \left\| X' \right\|^2 = \frac{1}{2} X^t Q'_3 X + \rho'^t_3 X + \epsilon_3 \quad (\text{III.42})$$

avec :

$$Q'_3 = \mathcal{I}, \quad \rho'_3 = 0 \quad (\text{III.43})$$

III.3.3.2 Formulation des contraintes

Lorsque le robot est en basculement sur deux roues, il est important de définir un certain nombre de contraintes :

- Respecter les contraintes de non-pénétration dans le sol, en assurant que l'angle de basculement Ψ est toujours positif ou nul.
- Respecter les limites physiques de la base mobile en limitant les vitesses et accélérations maximales
- Respecter la cinématique du corps du robot, en limitant la distance entre son CoM C^{xy} et celui de la base mobile B^{xy} .

Contrairement à la section (III.3.2.2), le robot ne peut pas tourner pendant le contrôle du basculement (B^t et C^t constant). Il n'est donc pas nécessaire de rendre les contraintes invariantes par rotation autour de la base B .

Dans les sections suivantes, nous écrirons les contraintes linéarisées de la façon suivante :

$$v'^{-}_i \leq V'_i X' \leq v'^{+}_i \quad (\text{III.44})$$

III.3.3.2.a Contraintes de non-pénétration dans le sol

La présence du sol fait que l'angle de basculement Φ ne peut pas être négatif. Cette contrainte est directement linéaire vis à vis des variables de commande et s'exprime ainsi :

$$\Phi \geq 0 \quad (\text{III.45})$$

On peut donc écrire la contrainte de la façon standard :

$$v'^{-}_1 \leq V'_1 X' \leq v'^{+}_1 \quad (\text{III.46})$$

avec :

$$\begin{aligned} V'_1 &= \begin{pmatrix} \tilde{U}^{-1}U_{\psi c} & \tilde{U}^{-1}U_{\psi b} \end{pmatrix} \\ v_1'^- &= \left(-\tilde{U}^{-1} \left(S_{\psi b}\hat{b}^n + S_{\psi c}\hat{c}^n + S_{\psi g^x}g^x + S_{\psi g^y}g^y + S_{\psi\psi}\hat{\psi} \right) \right) \\ v_1'^+ &= (+\infty) \end{aligned} \quad (\text{III.47})$$

III.3.3.2.b Vitesses et accélérations maximales de la base mobile

De façon identique à la section (III.3.2.2.b), les contraintes en vitesse et accélération de la base mobile s'écrivent de la façon suivante :

$$\left\| \vec{\dot{b}}_k \right\| \leq \dot{b}_m \quad (\text{III.48})$$

$$\left\| \vec{\ddot{b}}_k \right\| \leq \ddot{b}_m \quad (\text{III.49})$$

Le robot ne pouvant se déplacer qu'uniquement sur la droite représentée par le vecteur \vec{n}_k , les contraintes peuvent se réécrire de la façon suivante :

$$\dot{b}^n \leq \dot{b}_m \quad (\text{III.50})$$

$$\ddot{b}^n \leq \ddot{b}_m \quad (\text{III.51})$$

ce qui se réécrit de façon standard :

$$v_2'^- \leq V_2'X' \leq v_2'^+ \quad (\text{III.52})$$

avec :

$$V_2' = \begin{pmatrix} 0 & U_{\dot{b}} \\ 0 & U_{\ddot{b}} \end{pmatrix}, \quad v_2'^- = \begin{pmatrix} -\dot{b}_m - S_{\dot{b}}\hat{b}^n \\ -\ddot{b}_m - S_{\ddot{b}}\hat{b}^n \end{pmatrix}, \quad v_2'^+ = \begin{pmatrix} \dot{b}_m - S_{\dot{b}}\hat{b}^n \\ \ddot{b}_m - S_{\ddot{b}}\hat{b}^n \end{pmatrix} \quad (\text{III.53})$$

III.3.3.2.c Limites articulaires du corps du robot

De façon identique à la section (III.3.2.2.c), on définit la contrainte entre le CoM du corps du robot C et celui de la base mobile B par un rectangle de dimensions (k^x, k^y) centré en b .

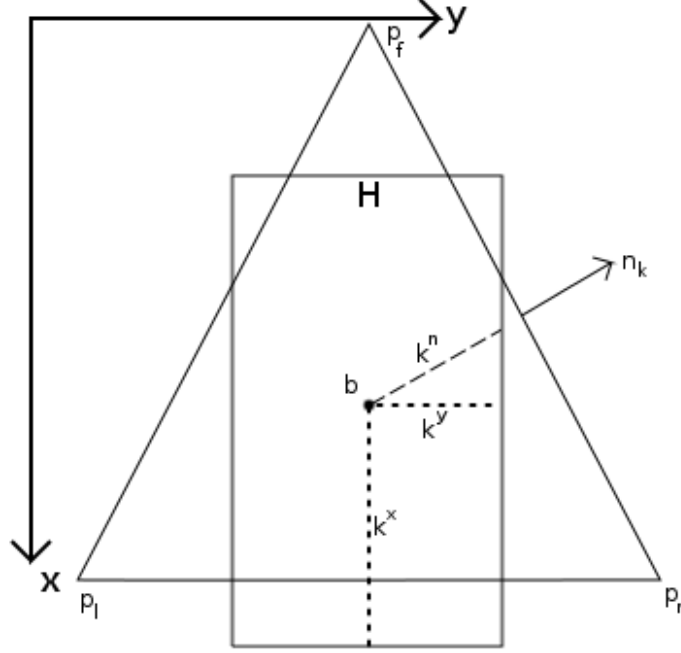


FIGURE III.3 – Représentation des contraintes cinématiques du corps du robot lorsque celui-ci bascule. H correspond à la contrainte exacte. k^n correspond à la distance entre le point b et le point d'intersection de l'axe (b, \vec{n}_k) et l'enveloppe H

Soit k^n la distance du segment reliant le centre de la base mobile B et le point d'intersection entre la droite définie par le vecteur \vec{n}_k et d'origine B avec le rectangle de la contrainte. k^n s'exprime de la façon suivante [Fig. III.3] :

$$k^n = \min \left(\left\| k^x \cos(\theta_k) \right\|, \left\| k^y \sin(\theta_k) \right\| \right) \quad (\text{III.54})$$

La contrainte s'exprime de la façon suivante :

$$-k^n \leq C^n - B^n \leq k^n \quad (\text{III.55})$$

ce qui se réécrit de façon standard :

$$v_3'^- \leq V_3' X' \leq v_3'^+ \quad (\text{III.56})$$

avec :

$$V'_3 = \begin{pmatrix} U_c & -U_b \end{pmatrix}, \quad v_3'^- = \begin{pmatrix} -k^n - S_c \hat{c}^n + S_b \hat{b}^n \end{pmatrix}, \quad v_3'^+ = \begin{pmatrix} k^n - S_c \hat{c}^n + S_b \hat{b}^n \end{pmatrix} \quad (\text{III.57})$$

III.3.3.3 Problème quadratique résultant

De la même manière qu'en section (III.3.2.3), nous pouvons écrire le problème quadratique résultant de la façon suivante :

$$\left\{ \begin{array}{l} \min_{X'} \left(X'^t \sum_{i=1}^3 \left(\alpha'_i Q'_i \right) X' + \sum_{i=1}^3 \left(\alpha'_i \rho_i'^t \right) X' \right) \\ \begin{pmatrix} v_1'^- \\ v_2'^- \\ v_3'^- \end{pmatrix} \leq \begin{pmatrix} V_1' \\ V_2' \\ V_3' \end{pmatrix} X \leq \begin{pmatrix} v_1'^+ \\ v_2'^+ \\ v_3'^+ \end{pmatrix} \end{array} \right. \quad (\text{III.58})$$

avec α'_i la pondération associée à l'objectif O'_i .

III.3.4 Gestion de la transition entre les deux états

Le contrôleur décrit en section (III.3.3) ne prend pas en compte la force de réaction qui se rajoute au moment où l'angle de basculement atteint 0. Cela amène ce contrôleur à générer une très grande accélération afin de maintenir l'angle à 0. Bien que celle-ci puisse être réalisable, ce comportement n'est pas souhaitable et nous préférons "laisser tomber" le robot sur le sol lorsque celui-ci va le percuter, afin d'éviter un déplacement exagéré de la base mobile.

Il faut donc adjoindre aux deux précédents contrôleurs un troisième permettant de gérer cet état. Nous choisissons d'utiliser le jeu de variable X . Les objectifs choisis sont :

- O_3 et O_4 . Contrôler le CoP au milieu de la base en utilisant le modèle de robot sur trois roues permet d'assurer au robot de ne pas accroître son angle de basculement. Cet objectif permet de contrôler le moment angulaire du robot autour de son axe de rotation afin qu'il soit toujours dirigé vers la bonne direction, lorsque celui-ci est proche du sol.
- O_5 . Assurer la stabilité numérique de l'algorithme est toujours indispensable.

Les contraintes choisies sont :

- (V_1, v_1^+, v_1^-) . La contrainte sur le CoP permet d'assurer au robot que son moment angulaire va faire tomber celui-ci sur ces trois roues, lorsque celui-ci est proche du sol.

- (V_2, v_2^+, v_2^-) et (V_3, v_3^+, v_3^-) . Afin de ne pas générer de mouvements infaisables, il faut contraindre le déplacement de la base mobile et du corps du robot.

Il faut cependant prendre en compte que le robot est toujours sous-actionné, car ayant une roue en l'air. Il s'impose donc de rajouter une contrainte concernant le déplacement de la base mobile, qui doit se faire uniquement selon l'axe \vec{n}_k . Cette contrainte s'exprime de la façon suivante :

$$\cos(\theta_k)\dot{B}^y = \sin(\theta_k)\dot{B}^x \quad (\text{III.59})$$

ce qui se réécrit de façon standard :

$$v_4^- \leq V_4 X \leq v_4^+ \quad (\text{III.60})$$

avec :

$$\begin{aligned} V_4 &= \begin{pmatrix} 0 & 0 & \sin(\theta_k)U_b & -\cos(\theta_k)U_b \end{pmatrix} \\ v_4^- = v_4^+ &= \begin{pmatrix} \cos(\theta_k)S_b\hat{b}^y - \sin(\theta_k)S_b\hat{b}^x \end{pmatrix} \end{aligned} \quad (\text{III.61})$$

Le problème quadratique résultant s'écrit donc :

$$\left\{ \begin{array}{l} \min_X \left(X^t \sum_{i=3}^5 (\alpha_i Q_i) X' + \sum_{i=3}^5 (\alpha_i \rho_i^t) X \right) \\ \begin{pmatrix} v_1^- \\ v_2^- \\ v_3^- \\ v_4^- \end{pmatrix} \leq \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} X \leq \begin{pmatrix} v_1^+ \\ v_2^+ \\ v_3^+ \\ v_4^+ \end{pmatrix} \end{array} \right. \quad (\text{III.62})$$

III.4 Gestion des deux modèles dynamiques exclusifs

III.4.1 Choix d'un superviseur et conséquences

L'objectif est maintenant de définir un moyen permettant de choisir quel contrôleur parmi les trois proposés en sections (III.3.2)(III.3.3)(III.3.4) doit être activé afin de commander le robot. Nous avons fait le choix d'un superviseur basé sur une machine à états et d'une heuristique de transition entre ces états [Fig. III.4].

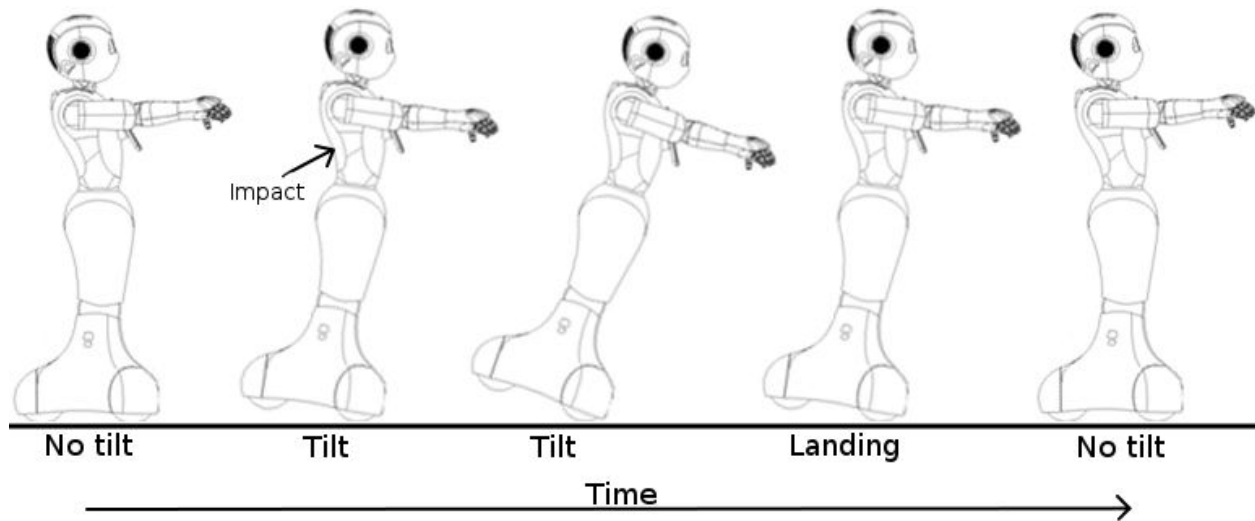


FIGURE III.4 – Présentation du changement des états du superviseur au cours du temps.

Les avantages de ce type de superviseur sont qu'il est facile à implémenter ainsi qu'à en faire évoluer les paramètres associés aux heuristiques de transition. De plus, ce type de superviseur permet de gérer de façon simple les transitions de modes dynamiques ainsi que les impacts.

Les inconvénients sont le fait que l'utilisation d'heuristiques rend le comportement du robot non-optimal à une situation donnée. De plus, si celles-ci sont mal choisies, son comportement peut être contraire aux objectifs de commande et impacter négativement son équilibre. Nous verrons notamment les conséquences de l'utilisation d'heuristiques de transitions au chapitre V

Dans la suite, il est important de prendre en compte le fait que certaines heuristiques ont été faites pour prendre en compte les perturbations les plus communes qu'un humain peut appliquer à la plate-forme expérimentale.

III.4.2 Fonctionnement du superviseur

III.4.2.1 Les états du superviseur

La machine à états du superviseur est composée de trois états [Fig. III.5] :

- L'état *No tilt* indique que le robot possède les trois roues sur le sol. Le contrôleur développé

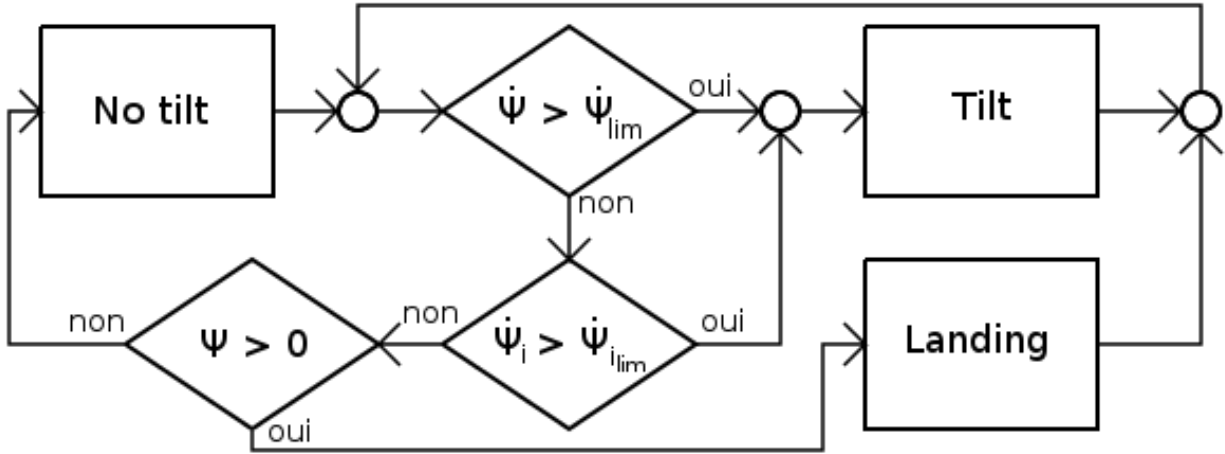


FIGURE III.5 – Schéma de fonctionnement du superviseur. Les boîtes “No tilt”, “Tilt” et “Landing” correspondent aux états du superviseur. $\dot{\psi}_i$ correspond à la future vitesse angulaire d’impact estimée et $\dot{\psi}_{lim}$ une limite à celle-ci. $\dot{\psi}$ correspond à la vitesse angulaire courante et $\dot{\psi}_{lim}$ une limite à celle-ci.

en section (III.3.2) est donc activé.

- L’état *Tilt* indique que le robot est en basculement sur deux roues. Le contrôleur développé en section (III.3.3) est donc activé.
- l’état *Landing* indique que le robot est en basculement sur deux roues, mais proche du sol et sur le point de le toucher. Le contrôleur développé en section (III.3.4) est donc activé.

Afin de définir les transitions entre les états, trois mesures sont utilisées : l’angle de basculement, la vitesse angulaire, et l’estimation de la vitesse angulaire d’impact.

III.4.2.2 Les transitions depuis l’état *No tilt*

Lorsque le superviseur est dans l’état *No tilt*, les transitions sont les suivantes [Fig. III.5] :

- Si la vitesse angulaire mesurée est supérieure à une limite donnée $\dot{\psi}_l$, alors le superviseur passe à l’état *Tilt*. On ne peut pas prévoir la durée ni l’intensité de la perturbation dans le futur. Ainsi, si on mesure une forte vitesse angulaire, on suppose que l’humain qui pousse

le robot ne va pas arrêter son geste rapidement, au moins pour des raisons d'inertie.

- Si la vitesse angulaire est inférieure à ψ_l , mais non-nulle, on utilise un estimateur de vitesse angulaire d'impact (qui sera décrit en section (III.4.3)). Si celui-ci indique que, sans compensation active par le contrôleur (III.3.3), le robot va tomber ou avoir une trop grande vitesse angulaire estimée au moment de l'impact, alors le superviseur passe à l'état *Tilt*. Cet estimateur permet de déterminer si le robot va tomber, en considérant un arrêt de la force de perturbation et une absence de compensation par le contrôleur (III.3.3). Si non, cet estimateur indique la vitesse angulaire d'impact.
- Dans le cas où l'estimateur indique que la vitesse angulaire d'impact estimée est suffisamment faible, alors le superviseur passe à l'état *Landing*. Cela signifie que le robot va compenser passivement la perturbation et ne pas générer un impact trop élevé. Dans ce cas, le mieux est de stopper le suivi de trajectoire, ainsi que de limiter le déplacement du robot dans l'axe de basculement, ce qui correspond bien au contrôleur de l'état *Landing*.
- Dans tous les autres cas, le superviseur reste dans l'état *No tilt*, car il n'est pas en train de basculer.

III.4.2.3 Les transitions depuis l'état *Tilt*

Lorsque le superviseur est dans l'état *Tilt*, les transitions sont les suivantes [Fig. III.5] :

- Si la vitesse angulaire mesurée est supérieure à ψ_l , alors le superviseur reste dans l'état *Tilt*. Cela signifie que la perturbation n'a toujours pas été contrôlée.
- Dans le cas contraire, on utilise l'estimateur afin de déterminer la vitesse angulaire d'impact. Si celle-ci est trop forte, alors le superviseur reste dans l'état *Tilt*.
- Dans le cas contraire, le superviseur passe à l'état *Landing*. Cela signifie que la perturbation a été suffisamment compensée pour à présent laisser tomber le robot sur le sol et finir de compenser passivement la perturbation.

III.4.2.4 Les transitions depuis l'état *Landing*

Lorsque le superviseur est dans l'état *Landing*, les transitions sont les suivantes [Fig. III.5] :

- Lorsque la vitesse angulaire et l'angle de basculement atteind 0, après une légère période de temporisation afin d'éviter tout rebond éventuel, le superviseur passe à l'état *No tilt*.

- Dans le cas contraire, on utilise l'estimateur afin de déterminer la vitesse angulaire d'impact. Si celle-ci est trop grande, alors le superviseur passe à l'état *Tilt*.
- Dans le cas contraire, alors la phase atterrissage n'est pas terminée et le superviseur reste dans l'état *Landing*.

III.4.3 Fonctionnement de l'estimateur d'impact

Le premier objectif de l'estimateur d'impact est de déterminer si le robot va pouvoir compenser passivement la perturbation, ou bien tomber. Le second objectif est, dans le cas où le robot va pouvoir compenser passivement la perturbation, estimer la vitesse angulaire au moment de l'impact.

L'estimateur utilise l'observation courante de l'angle de basculement ainsi que la mesure de la vitesse angulaire. On suppose dans un premier temps que le robot a une accélération angulaire constante dépendant uniquement de la gravité et de l'angle actuel de basculement :

$$\ddot{\psi} = -\frac{g^z}{h_b} \cos\left(\psi_0 + \text{atan}\left(\frac{h_b}{d_k}\right)\right) \quad (\text{III.63})$$

A l'aide de l'équation (III.63), on peut écrire l'évolution de l'angle de basculement au cours du temps :

$$\psi = -\frac{g^z \cos\left(\psi_0 + \text{atan}\left(\frac{h_b}{d_k}\right)\right)}{2h_b} t^2 + \dot{\psi}_0 t + \psi_0 \quad (\text{III.64})$$

En résolvant l'équation quadratique associée, pour $\psi = 0$, on peut déterminer si le robot va impacter le sol, et si oui, estimer quel va être la vitesse angulaire d'impact. Soit Δ tel que :

$$\Delta = \dot{\psi}_0^2 + \frac{2g^z \psi_0 \cos\left(\psi_0 + \text{atan}\left(\frac{h_b}{d_k}\right)\right)}{h_b} \quad (\text{III.65})$$

Si $\Delta < 0$, le robot va tomber et il n'y aura pas d'impact de la roue en l'air sur le sol. Dans le cas contraire, le temps d'impact t_i est calculé de la manière suivante :

$$t_i = \frac{h_b(\dot{\psi}_0 \pm \sqrt{\Delta})}{g^z \cos\left(\psi_0 + \text{atan}\left(\frac{h_b}{d_k}\right)\right)} \quad (\text{III.66})$$

et la vitesse d'impact estimée $\dot{\psi}_i$ est :

$$\dot{\psi}_i = -\sqrt{\Delta} \quad (\text{III.67})$$

III.5 Vers une modélisation unifiée des deux dynamiques

Dans cette section, nous présentons une méthode de résolution du problème d'unification des deux modèles dynamiques (II.122)(II.134). nous n'avons pas eu le temps d'implémenter cette méthode sur la plate-forme expérimentale, nous allons donc apporter uniquement un début d'approche théorique.

Afin d'unifier les deux modèles dynamiques en un seul contrôleur, tout en restant dans le formalisme de la programmation quadratique linéaire, une solution est de résoudre *a priori* le problème d'optimisation sur l'ensemble de l'horizon de prédiction. Nous proposons l'heuristique suivante :

- Dans le cas où le robot possède initialement les trois roues en contact avec le sol, on considère que celui-ci le restera durant tout l'horizon de prédiction. La loi de commande utilisée est directement celle présentée en section (III.3.2).
- Dans le cas où le robot est initialement en basculement sur deux roues (suite à une forte perturbation par exemple), on considère que celui-ci le restera sur l'intervalle $[0, m]$ de l'horizon de prédiction, où $m \leq n$. La loi de commande utilisée sur cet intervalle est de la forme de celle présentée en section (III.3.3) Sur l'intervalle $[m + 1, n]$, si celui-ci existe ($m < n$), on considère que le robot est revenue sur ses trois roues. La loi de commande utilisée sur cet intervalle est de la forme de celle présentée en section (III.3.2).

Cette heuristique implique notamment le fait que lorsque le robot est en basculement sur deux roues, celui-ci va soit le rester durant tout l'horizon, soit retomber sans rebondir à l'instant m . Cela est théoriquement pris en compte par les contraintes sur le CoP sur l'intervalle $[m + 1, n]$, assurant au robot de ne pas rebondir. Il faut pour cela intégrer dans la formulation du CoP sur les instants $[m + 1, n]$ l'influence de la vitesse et de l'accélération angulaire d'impact sur la formulation du CoP.

De plus, une prédiction de la valeur de m doit être faite. Il faut donc, en s'inspirant de l'estimateur d'impact par exemple (III.4.3), prédire cet instant m et s'assurer qu'il converge vers la véritable valeur au fur et à mesure que m tend vers 0.

Enfin, un travail de réécriture de la dynamique future doit être faite, de façon analogue à ce qui a été fait en section (II.3), afin de prendre en compte les deux modèles, ainsi que de leur transition entre les instants m et $m + 1$.

III.6 Conclusion

Dans ce chapitre, nous avons présenté une loi de commande composite permettant de contrôler l'équilibre et les mouvements d'un robot humanoïde à roues omnidirectionnelles. Celle-ci se base sur trois sous-lois de commandes optimales utilisant le formalisme de la programmation quadratique et supervisées par une machine à état. Ces trois lois permettent de contrôler le robot dans des situations différentes :

- Lorsque le robot est en contact sur le sol avec ses trois roues.
- Lorsque le robot bascule sur deux roues.
- Lorsque le robot va impacter le sol.

En estimant l'existence et la vitesse angulaire future d'impact du robot avec le sol, ainsi qu'en mesurant l'angle de basculement courant et sa vitesse angulaire, le superviseur est capable de choisir quelle sous-loi de commande activer à chaque instant.

Enfin, l'ébauche d'une approche unifiée de cette loi de commande a été présentée afin de se passer de l'utilisation d'un superviseur, ainsi que d'améliorer la gestion de la transition entre les deux modes dynamiques (trois ou deux roues en contact avec le sol).

Chapitre IV

Mesures et observateurs

IV.1 Introduction

L'objectif de ce chapitre est de présenter les différentes mesures et observateurs qui sont nécessaires à la réalisation de la commande présentée en chapitre III. Nous présenterons dans un premier temps les différentes valeurs à observer, puis la liste des capteurs disponibles sur la plate-forme expérimentale. Ensuite de quoi, nous détaillerons les méthodes de mesures ainsi que les observateurs permettant de réaliser cela. Enfin, nous discuterons des conséquences de chaque méthode sur la stabilité et le comportement final du robot.

Les observateurs présentés dans ce chapitre ont été implémentés directement sur la plate-forme expérimentale. Les performances obtenues sur le robot n'étant pas limités par ceux-ci, ils ont volontairement été réalisés de façon simple. Le contenu de ce chapitre sera donc court, et ne traitera pas en détail de problématiques comme les bruits capteurs ou l'impact des incertitudes du modèle du robot sur les observations.

IV.2 Valeurs à observer et capteurs disponibles

La loi de commande présentée en chapitre III nécessite plusieurs variables d'entrées :

- L'état complet du CoM du corps du robot \hat{c} .
- L'état complet du CoM de la base mobile \hat{b} .

- L'angle de basculement et la vitesse angulaire, nécessaires au fonctionnement du superviseur (III.4).
- L'orientation du sol, nécessaire au calcul du vecteur gravité g .

Afin de mesurer ces valeurs, la plate-forme expérimentale dispose de plusieurs capteurs :

- Associé à chaque moteur, des capteurs magnéto-résistifs permettent de mesurer son angle ou sa vitesse angulaire. Ces capteurs servent à mesurer la vitesse angulaire des roues de la base mobile. Sur chacune des articulation du robot, ces capteurs servent à mesurer l'angle de l'articulation.
- Une centrale inertielle, disposée au centre de la base mobile, permet de mesurer l'accélération linéaire ainsi que la vitesse de rotation de la base mobile.

Nous pouvons constater dans un premier temps qu'aucune des variables d'entrée de la loi de commande n'est directement mesurée par les capteurs. Nous allons devoir réaliser des observateurs, afin d'estimer au mieux possibles ces valeurs.

IV.3 Méthodes de mesure et conséquences

IV.3.1 Observation de l'état du CoM de la base mobile

IV.3.1.1 Vitesse du CoM de la base mobile

Les capteurs magnéto-résistifs associés aux moteurs des roues du robots nous permettent de mesurer la vitesse de rotation de celles-ci. En utilisant le modèle géométrique de la base mobile, nous pouvons en déduire directement une vitesse de déplacement de la base mobile. Cette déduction est valable uniquement si les roues ne glissent pas sur le sol.

Soit β_i la vitesse de la roue $i \in \{f, r, l\}$ et W une matrice de transformation utilisant le modèle géométrique de la base mobile pour exprimer la vitesse de déplacement de la base mobile en fonction de celles des roues :

$$\begin{pmatrix} \dot{b}^x \\ \dot{b}^y \end{pmatrix} = W \begin{pmatrix} \beta_f \\ \beta_r \\ \beta_l \end{pmatrix} \quad (\text{IV.1})$$

Les différences entre la vitesse réelle de la base mobile et celle observée proviennent directement des glissements entre les roues et le sol, dépendants de la nature du sol et de l'usure du revêtement des roues.

Une méthode, qui n'a pas été implémentée, afin de mesurer ce glissement serait d'utiliser les accéléromètres de la centrale inertielle de la base mobile.

IV.3.1.2 Position du CoM de la base mobile

La position de la base mobile est directement obtenue par intégration des vitesses mesurées :

$$\begin{pmatrix} b_\tau^x \\ b_\tau^y \end{pmatrix} = \begin{pmatrix} b_0^x \\ b_0^y \end{pmatrix} + \tau \begin{pmatrix} \dot{b}_0^x \\ \dot{b}_0^y \end{pmatrix} \quad (\text{IV.2})$$

avec τ la période d'échantillonnage de la mesure des β_i .

Le défaut de cette approche est une dérive de la position du robot dans le temps. Afin de compenser cette dérive lorsque le robot est immobile, principalement à cause du bruit capteur, un filtre en amplitude a été ajouté, ignorant les mesures de faible amplitude.

IV.3.1.3 Accélération du CoM de la base mobile

Concernant l'observation de l'accélération de la base mobile, le bruit de quantification apporté par une dérivation d'Euler de la vitesse mesurée ne nous permet pas d'obtenir une valeur utilisable. La solution qui a été choisie est d'utiliser l'équation d'état (II.103) conjointement à la sortie de la loi de commande X calculé au pas de temps précédent :

$$\ddot{b}_\tau^x = \tau \ddot{B}(0) + \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \hat{b}_0 \quad (\text{IV.3})$$

où $*(0)$ correspond à la première valeur du vecteur $*$.

Si la commande $\ddot{B}(0)$ est bien réalisée par les moteurs, et en l'absence de perturbations, alors l'accélération observée \ddot{b}_τ^x correspond exactement à la valeur réelle. Les informations des erreurs générées par les perturbations vont cependant arriver avec un certain retard, dépendant des observations aux instants précédents de b^x et \dot{b}^x .

On peut noter que les accéléromètres présents dans la base mobile ne permettent pas de mesurer correctement \ddot{b} car ceux-ci sont trop bruités.

IV.3.2 Observation de l'état du CoM du corps du robot

IV.3.2.1 Position du CoM du corps du robot

Le corps du robot étant mécaniquement lié à la base mobile, on peut composer la position de son CoM en utilisant sa posture et la position du CoM de la base mobile b :

$$c = b + \delta_c \quad (\text{IV.4})$$

Où $\delta_c = c - b$, correspond aux informations apportées par la posture du robot.

En utilisant le modèle géométrique du robot, ainsi que les capteurs de position articulaire, nous pouvons directement mesurer la valeur δ_c . Cette mesure est valable tant que le modèle géométrique correspond au corps du robot. Ainsi, les déformations pouvant apparaître dans la structure du robot ne sont pas prises en compte.

IV.3.2.2 Vitesse et accélération du CoM du corps du robot

Nous ne disposons d'aucune mesure de vitesse et d'accélération. De plus, dériver directement la position du robot c apporte beaucoup trop de bruit de quantification. Nous allons procéder de la même manière qu'en section (IV.3.1.3), en utilisant l'équation d'état (II.103) et la commande X :

$$\dot{c}_\tau^x = \frac{\tau^2}{2} \ddot{C}(0) + \begin{pmatrix} 0 & 1 & \tau \end{pmatrix} \hat{c}_0 \quad (\text{IV.5})$$

$$\ddot{c}_\tau^x = \tau \ddot{C}(0) + \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \hat{c}_0 \quad (\text{IV.6})$$

IV.3.3 Observation de l'angle de basculement et d'inclinaison du sol

Le capteur disponible permettant d'observer l'angle de basculement et l'inclinaison du sol est la centrale inertielle disposée dans la base mobile. Le problème principal étant que ce capteur ne peut mesurer que la somme des deux angles et vitesses angulaires. Sans utilisation d'*a priori*, le système est donc non observable.

Dans le chapitre II, nous avons fait l'hypothèse d'une inclinaison du sol constante. Afin de rendre le système observable, nous allons donc définir les *a priori* suivants :

- Lorsque la vitesse angulaire mesurée est nulle durant un temps minimum défini, on considère l’angle de basculement nul également. L’angle mesuré par la centrale inertielle correspond alors à l’inclinaison du sol.
- Lorsque la vitesse angulaire mesurée est non-nulle durant un temps minimum défini, on considère l’inclinaison du sol constante. L’angle et la vitesse angulaire mesurés correspondent donc au basculement du robot.

Deux cas limites émergent de ces *a priori* :

- A cause du bruit capteur sur les gyromètres, si l’on amène le robot à basculer suffisamment lentement afin que la vitesse angulaire soit inférieure au bruit, alors il n’est pas possible de savoir que le robot est en train de basculer. Cela peut être résolu par un capteur de force sur chaque roue, permettant de connaître exactement quelle roue est en l’air, et lesquelles sont au sol. Dans notre cas, on suppose que cela n’arrive pas.
- Pendant le moment où le robot monte une pente, celle-ci est confondue avec un basculement, ce qui peut générer une commande non correcte du robot (accélération de la base mobile dans le sens descendant de la pente). Cela est résolue par la temporisation minimum à partir duquel le robot change d’a priori, en considérant que le profil spatial d’inclinaison du sol est à variation nulle par morceaux (les changements d’inclinaison du sol sont spatialement instantanés, ce qui est le cas de la majorité des pentes que l’on peut croiser dans l’environnement urbain).

Mis à part ces deux cas limites, l’utilisation de ces *a priori* permet d’observer simultanément l’angle de basculement et l’inclinaison du sol. Concernant la vitesse angulaire, celle-ci est directement extraite à partir des gyromètres, préalablement filtrés. Quant à l’angle de basculement/d’inclinaison, celui-ci est calculé en utilisant un algorithme présenté dans l’article [30]

Il est possible de se passer de ces *a priori* en utilisant des capteurs supplémentaires afin d’observer simultanément l’angle de basculement et celui d’inclinaison du sol. On peut trouver différents travaux utilisant une caméra [33, 47], un laser [42], ou encore un capteur de force [41], afin de mesurer l’angle entre le pied du robot et le sol. En utilisant ces travaux, couplés avec une mesure de la somme des deux angles via une centrale inertielle, il est possible de les observer simultanément sans *a priori*.

Chapitre V

Résultats et expérimentations

V.1 Introduction

Au cours de ce chapitre, nous présenterons dans un premier temps le schéma de contrôle en boucle fermée implémenté sur la plate-forme expérimentale. Nous présenterons ensuite plusieurs résultats de la loi de commande développée au chapitre III. Chaque expérience permettra de mettre en évidence un aspect de la loi de commande.

V.2 Schéma de contrôle en boucle fermée

L'architecture de commande de la plate-forme expérimentale est séparée sur plusieurs processeurs : pour chaque actionneur, un micro-contrôleur permet de le commander. Un processeur principal permet de réaliser la loi de commande présentée au chapitre III.

Dans les sections suivantes, nous détaillerons les différents éléments du schéma de contrôle en boucle fermée [Fig. V.1].

V.2.1 Boucle de commande moteur

Les micro-contrôleurs commandant chaque actionneur sont échantillonnés à 1 kHz. Ils implémentent une boucle de commande de type Proportionnel-Intégral (PI). Ces contrôles sont en

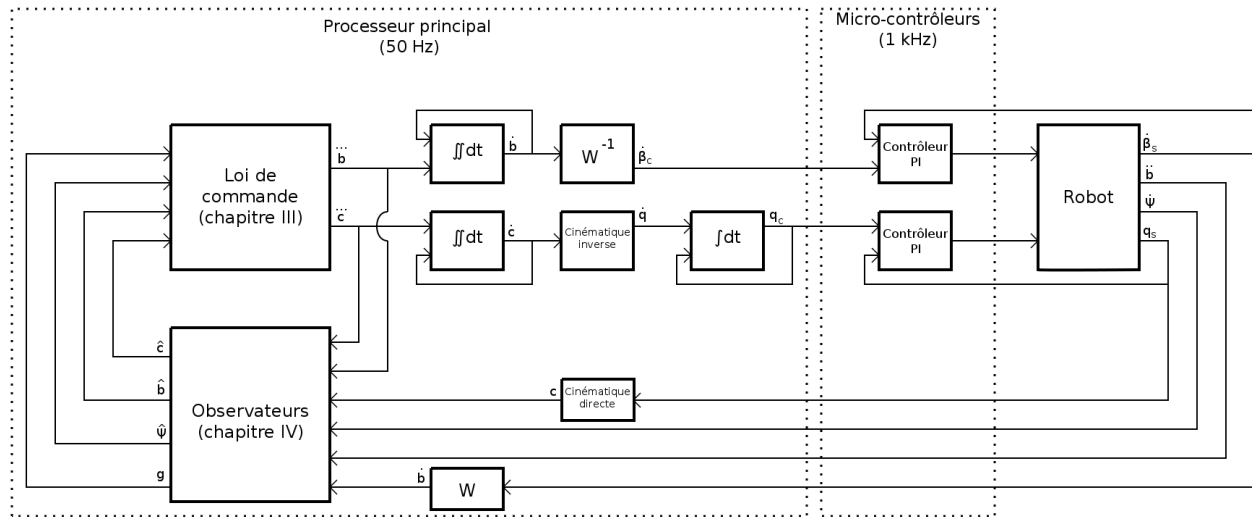


FIGURE V.1 – Schéma de contrôle en boucle fermée.

position pour les articulations du corps du robot, et en vitesse pour les roues de la base mobile. Les micro-contrôleurs communiquent avec le processeur principal afin de récupérer les références en position ou vitesse, puis génèrent une commande envoyée directement aux actionneurs.

Ces asservissements étant fournis avec la plate-forme expérimentale, il n'a pas été possible de les modifier. On peut noter que le comportement du robot pourrait être significativement amélioré avec un contrôleur prenant en compte l'accélération et le couple moteur.

V.2.2 Boucle de commande principale

La boucle de commande principale est échantillonnée à 50 Hz. Elle est séparée en trois éléments :

- Le premier a pour entrée les données capteurs (vitesses des roues, accélération de la base mobile, angle de basculement et positions articulaires). Les positions articulaires sont converties en position du centre de masse du corps du robot via un calcul standard de géométrie directe, utilisant un modèle du robot. La vitesse de la base mobile est obtenue en multipliant le vecteur des vitesses des roues par la matrice W (IV.1).
- Le second élément correspond aux observateurs présentés au chapitre IV suivis de la loi de

commande du chapitre III. Il a pour entrée la sortie du premier élément décrit précédemment ainsi que les données de la centrale inertielle de la base mobile (accélération de la base et vitesse angulaire). En sortie est obtenu les jerks de commande des CoMs de la base mobile et du corps du robot.

- Le dernier élément permet de convertir les jerks de commande calculés précédemment en commandes de position de chaque articulation et de vitesse des roues de la base mobile. Les positions articulaires de commande sont obtenues en intégrant le jerk du CoM du corps du robot puis en résolvant un problème standard de cinématique inverse et en intégrant le résultat. Pour information, le problème de cinématique inverse est résolu en utilisant la librairie METAPOD [29]. Les vitesses des roues sont obtenues en intégrant le jerk du CoM de la base mobile puis en multipliant le résultat par la matrice W^{-1} .

V.2.3 Filtrages capteur

Quelques filtrages supplémentaires sont nécessaires afin de compenser divers éléments parasites non pris en compte par la loi de commande principale :

- Les mesures des positions articulaires et vitesses capteurs sont quantifiées. Ainsi, afin de minimiser le bruit de quantification, lorsque la variation entre la mesure actuelle et celle au pas de temps précédent est de l'ordre de grandeur du pas de quantification, alors cette variation est ignorée. Il existe des méthodes plus sophistiquées afin de compenser le bruit de quantification [28], mais il n'a pas été pertinent de les implémenter, principalement pour des raisons de temps de développement ainsi que de besoins industriels.
- Le temps de communication entre les capteurs et le processeur principal n'étant pas nul, en ajoutant le retard capteur, les mesures obtenues dans le processeur principal ont un délai conséquent. Afin de le compenser, on utilise un prédictor de Smith [6] sous la forme suivante [Fig. V.2] :

$$\tilde{q}_{s_0} = q_0 + (q_{s-\delta} - q_{-\delta}) \quad (\text{V.1})$$

$$(\text{V.2})$$

avec \tilde{q}_{s_0} la mesure estimée à l'instant courant, $q_{s-\delta}$ la mesure retardée de δ secondes, $q_{-\delta}$ la commande envoyée δ secondes avant et q_0 la commande courante.

Ce filtrage suppose que l'erreur entre la commande et la mesure à un moment donné varie peu sur la période δ . Ainsi, en connaissant cette erreur, on peut estimer la valeur de la mesure courante.

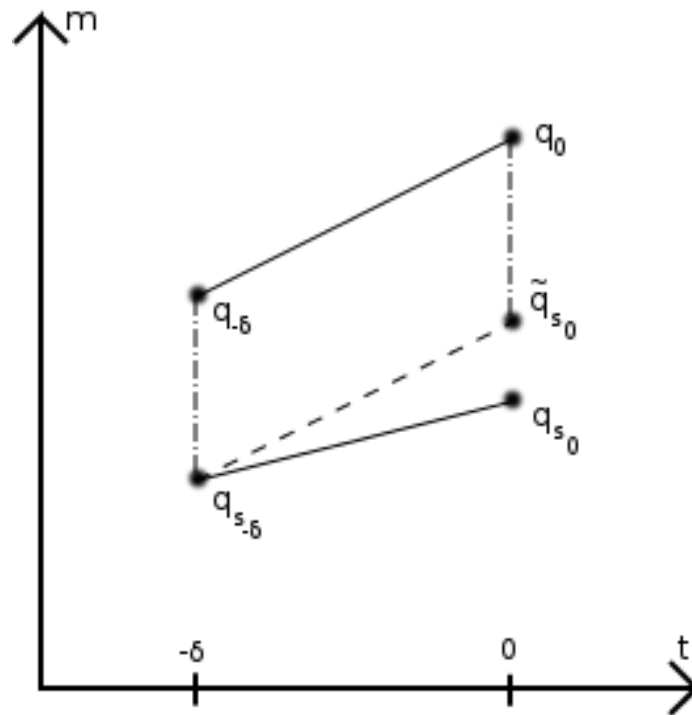


FIGURE V.2 – Illustration du filtre de compensation du retard capteur.

V.3 Expériences

V.3.1 Introduction

La loi de commande a été implémentée sur l'ensemble du parc de robots Pepper produits par Aldebaran. Au cours de cette thèse, un temps important a porté sur l'implémentation et la qualité des résultats en situation réelle [Fig. V.3][Fig. V.4]. Notamment, de nombreux essais ont été fait concernant la compensation du basculement lorsqu'un client déséquilibre le robot en le poussant.

Afin de montrer le comportement de la loi de commande de façon claire, nous avons choisi de ne présenter dans cette section qu'un nombre réduit d'expériences. Celles-ci vont permettre de mettre en évidence les différents aspects de la loi de commande.



FIGURE V.3 – Illustration d'un robot Pepper dans un magasin Softbank Mobile au Japon.



FIGURE V.4 – Illustration d'un robot Pepper en interaction dans un supermarché Japonais.

V.3.2 Suivi de trajectoires non réalisables

V.3.2.1 Protocole expérimental

L'objectif de cette première expérience [Fig. V.5][Fig. V.6] est de montrer l'influence du choix des pondérations dans le comportement de la loi de commande.

Le protocole expérimental est le suivant : on dispose le robot les trois roues sur un sol plat et horizontal, en donnant en entrée de la loi de commande une trajectoire de référence non réalisable, avec une discontinuité en vitesse à mi-chemin [Fig. V.5][Fig. V.6]. La trajectoire consiste en un aller-retour latéral de 70cm. Celle-ci est dérivable deux fois, excepté à $t = 2.6s$ où une discontinuité en vitesse a été ajoutée.

V.3.2.2 Analyse

Sur la figure [Fig. V.5], le jeu de pondérations choisi pour la loi de commande privilégie le suivi de trajectoire vis à vis de la robustesse aux perturbations. Il en résulte un suivi de trajectoire

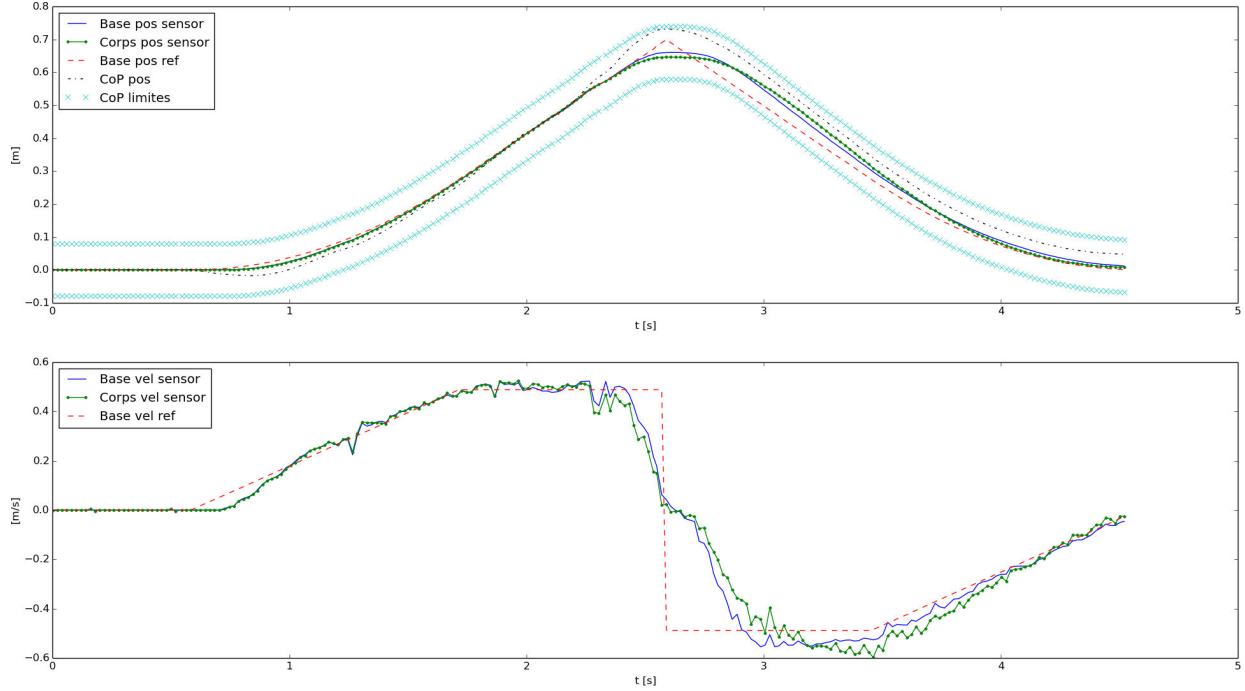


FIGURE V.5 – Suivi d’une trajectoire non-réalisable avec un jeu de pondérations priorisant le suivi de trajectoire par rapport à la robustesse.

précis. Du fait de l’aspect prédictif de la loi de commande, on peut observer une anticipation de la discontinuité, par une courbure de la trajectoire (et donc une augmentation de l’erreur de suivi instantané) en avance par rapport à la discontinuité. Cette anticipation a pour but de minimiser l’erreur sur l’ensemble de la trajectoire.

Le délai de démarrage du mouvement du robot, observé à $t = 0.7s$, est dû aux frottements secs situé dans la chaîne de transmission entre les moteurs des roues et le sol.

On peut noter également que le CoP du robot est proche de ses limites lors de la discontinuité, cela a pour but d’assurer un meilleur suivi de trajectoire, au détriment de la robustesse.

Concernant la figure [Fig. V.6], le jeu de pondérations choisi pour la loi de commande privilégie la robustesse aux perturbations vis à vis du suivi de trajectoires. Il en résulte un suivi de trajectoire médiocre, mais une distance entre le centre du robot (position de la base) et le CoP plus faible, ce qui améliore la robustesse.

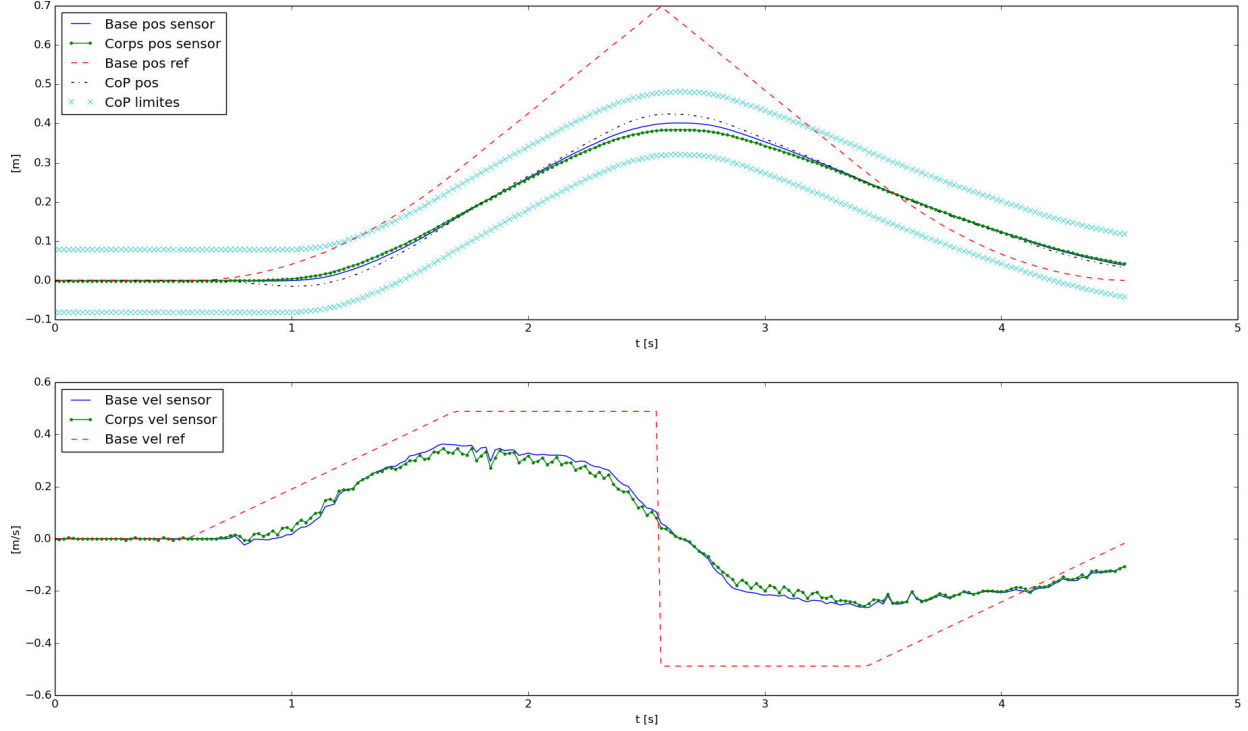


FIGURE V.6 – Suivi d’une trajectoire non-réalisable avec un jeu de pondérations priorisant la robustesse par rapport au suivi de trajectoire.

V.3.3 Suivi de trajectoire en présence de faibles perturbations

V.3.3.1 Protocole expérimental

L’objectif de cette seconde expérience [Fig. V.7] est de montrer la capacité de la loi de commande à compenser une faible perturbation de la base mobile.

Le protocole expérimental est le suivant : On dispose le robot les trois roues sur un sol plat et horizontal, en donnant en entrée de la loi de commande une trajectoire de référence dérivable deux fois. Celle-ci consiste en un déplacement de 2.8m vers l’avant. A $t = 2.3s$, on perturbe le robot en poussant la base mobile vers l’arrière pendant environ une seconde.

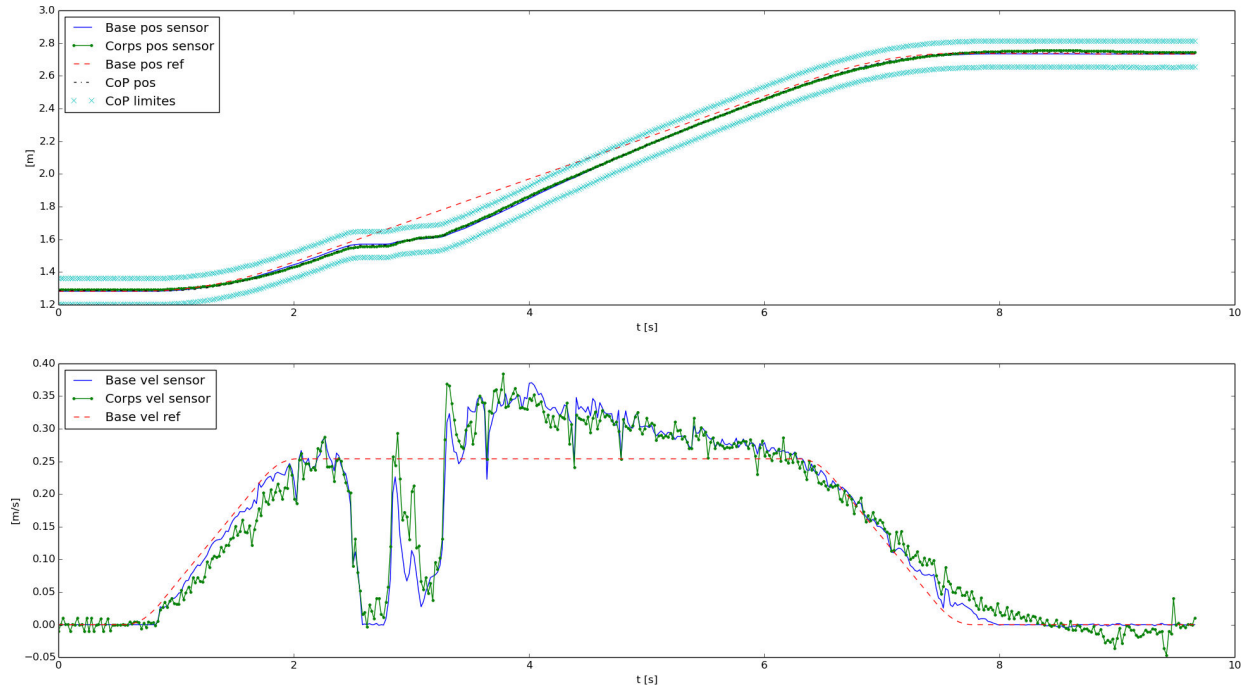


FIGURE V.7 – Suivi d’une trajectoire en présence d’une faible perturbation de la base mobile.

V.3.3.2 Analyse

Sur la figure [Fig. V.7], on peut dans un premier temps observer la perturbation occasionnée au robot, qui impose une vitesse de déplacement autour de 0 pendant une seconde environ. Ensuite, afin de rattraper le retard en suivi de position, la loi de commande génère une erreur en suivi de vitesse, en se déplaçant plus vite que désiré. La perturbation est compensée à la fois en vitesse et en position en trois secondes environs.

L’écart relatif entre les pondérations de suivi en position et en vitesse conditionne le temps de compensation de la perturbation : si la pondération en suivi de position est très supérieure à celle en vitesse, alors la perturbation sera vite rattrapée. Si le suivi en vitesse est priorisé par rapport au suivi en position, alors le robot compensera l’erreur en position beaucoup plus lentement.

V.3.4 Compensation du basculement du robot

V.3.4.1 Lorsque le robot est immobile

V.3.4.1.a Protocole expérimental

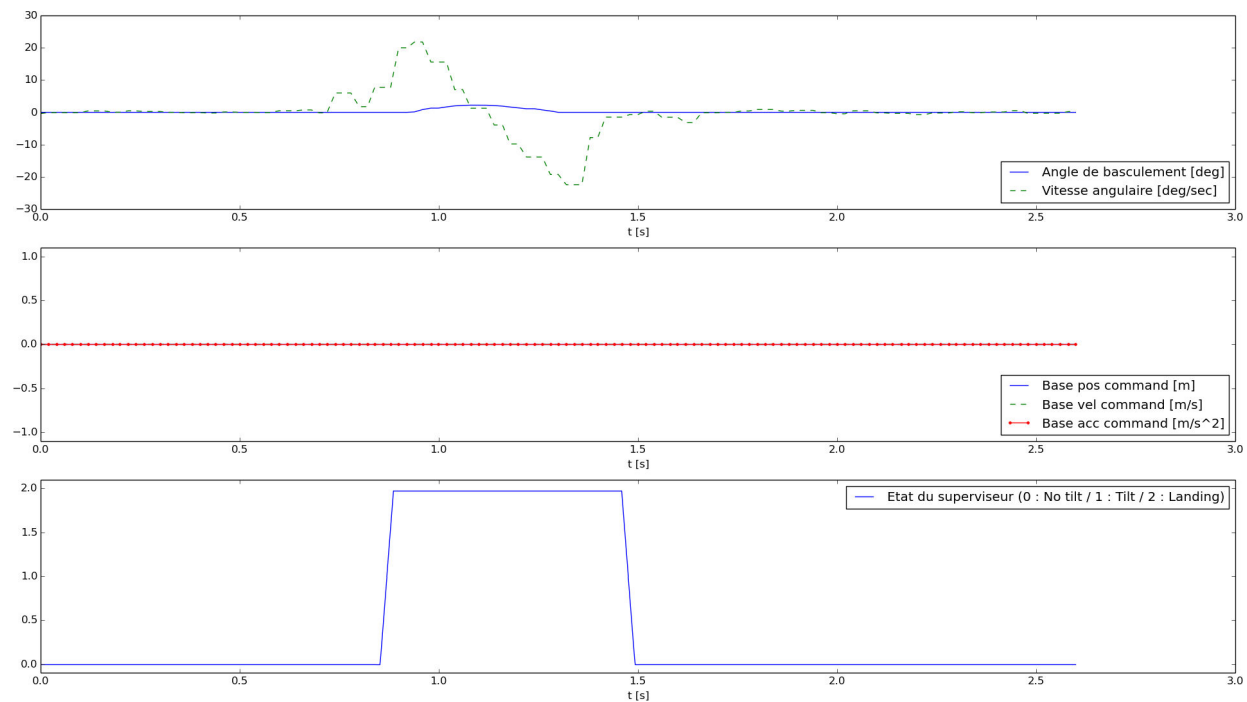


FIGURE V.8 – Compensation d'un basculement du robot en présence d'une faible perturbation.

L'objectif de cette troisième expérience [Fig. V.8][Fig. V.9][Fig. V.10] est de présenter le comportement ainsi que les performances de la loi de commande lorsque l'on perturbe le robot de façon à le faire basculer sur deux roues. Dans un premier temps, nous présentons le cas où le robot est initialement immobile.

Le protocole expérimental est le suivant : on dispose le robot les trois roues sur un sol plat et horizontal, puis on le pousse par les épaules afin de générer un basculement plus ou moins fort.

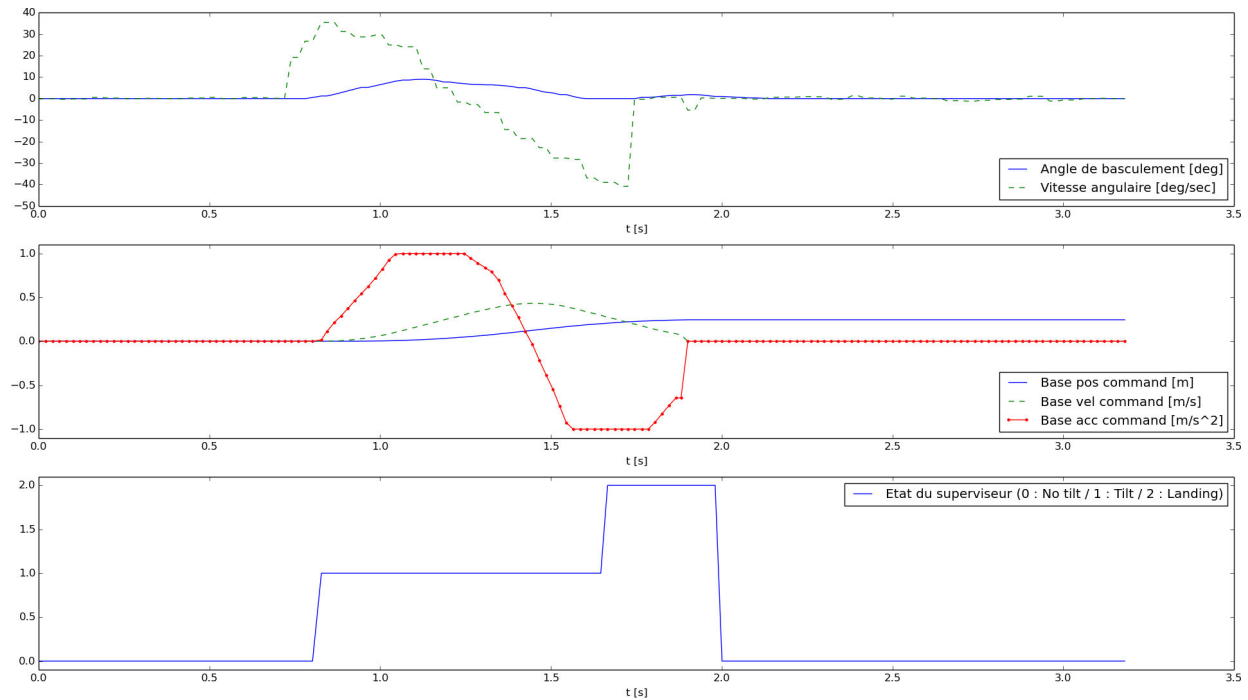


FIGURE V.9 – Compensation d’un basculement du robot en présence d’une perturbation moyenne.

V.3.4.1.b Analyse

La figure [Fig. V.8] montre l’exemple d’une faible perturbation. Le superviseur possède un seuil limite de vitesse d’impact estimée d’environ 40 deg/s. Celle-ci n’ayant pas été dépassée, la compensation active de basculement n’a pas été déclenchée. La loi de commande a donc laissé la gravité faire atterrir le robot à environ 25 deg/s sur le sol.

La figure [Fig. V.9] montre l’exemple d’une perturbation moyenne. Dans ce cas, la poussée génère beaucoup de vitesse angulaire, et le superviseur déclenche la compensation active à environ 0.8s. On observe ensuite des mouvements de la base mobile ayant pour objectif de réduire l’angle de basculement ainsi que la vitesse angulaire au moment de l’impact.

La perturbation est compensée en 1s environ. Il est à noter que l’accélération de la base mobile est limitée à $1m/s^2$. Si celle-ci était plus grande, les performances seraient meilleures.

La figure [Fig. V.10] montre l’exemple d’une perturbation trop forte que la loi de commande

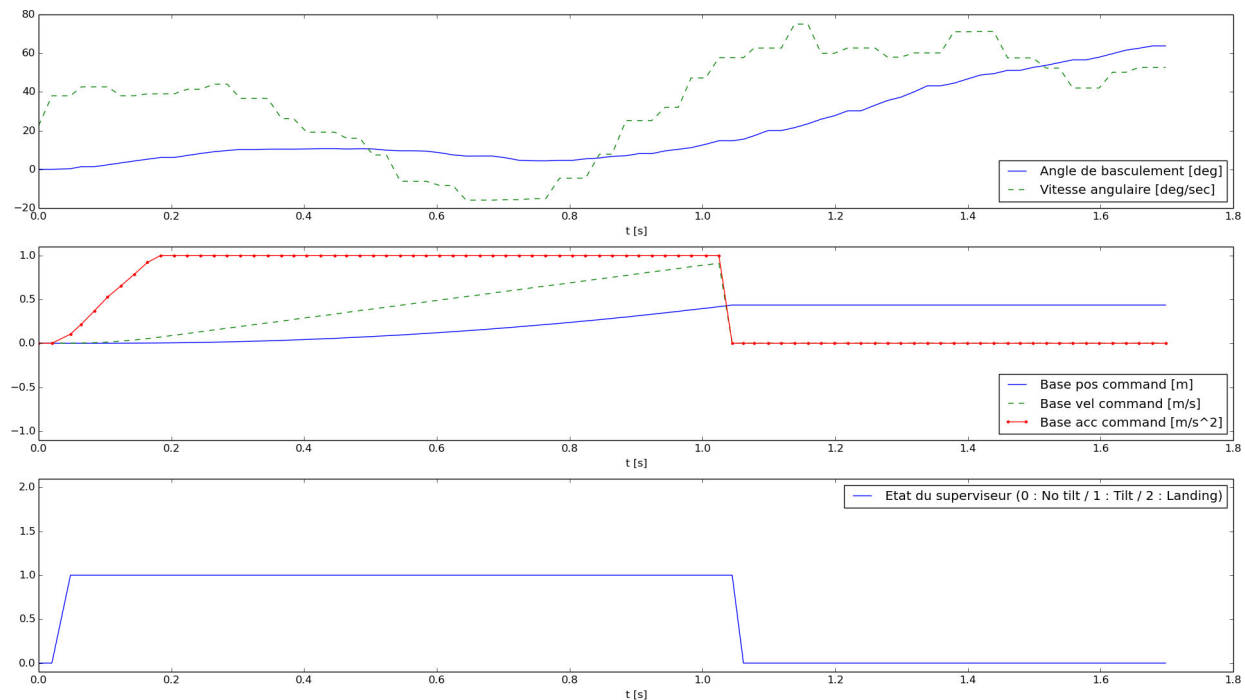


FIGURE V.10 – Compensation d’un basculement du robot en présence d’une forte perturbation.

est incapable de compenser. Le jerk de commande de la base mobile est saturé dès le début de la compensation (progression linéaire de l’accélération), puis à 0.2s l’accélération sature à son tour. Avec les limitations actuelles des moteurs de la base mobile, il n’est pas possible de rééquilibrer le robot. Une stratégie de protection contre la chute se déclenche sur le robot à environs 1.05s, arrêtant toutes les commandes envoyées.

V.3.4.2 Lorsque le robot est en mouvement

V.3.4.2.a Protocole expérimental

L’objectif de cette quatrième expérience [Fig. V.11] est de présenter le comportement ainsi que les performances de la loi de commande lorsque l’on perturbe le robot de façon à le faire basculer sur deux roues. Nous présentons ici le cas où le robot est initialement en mouvement.

Le protocole expérimental est le suivant : on dispose le robot les trois roues sur un sol plat et

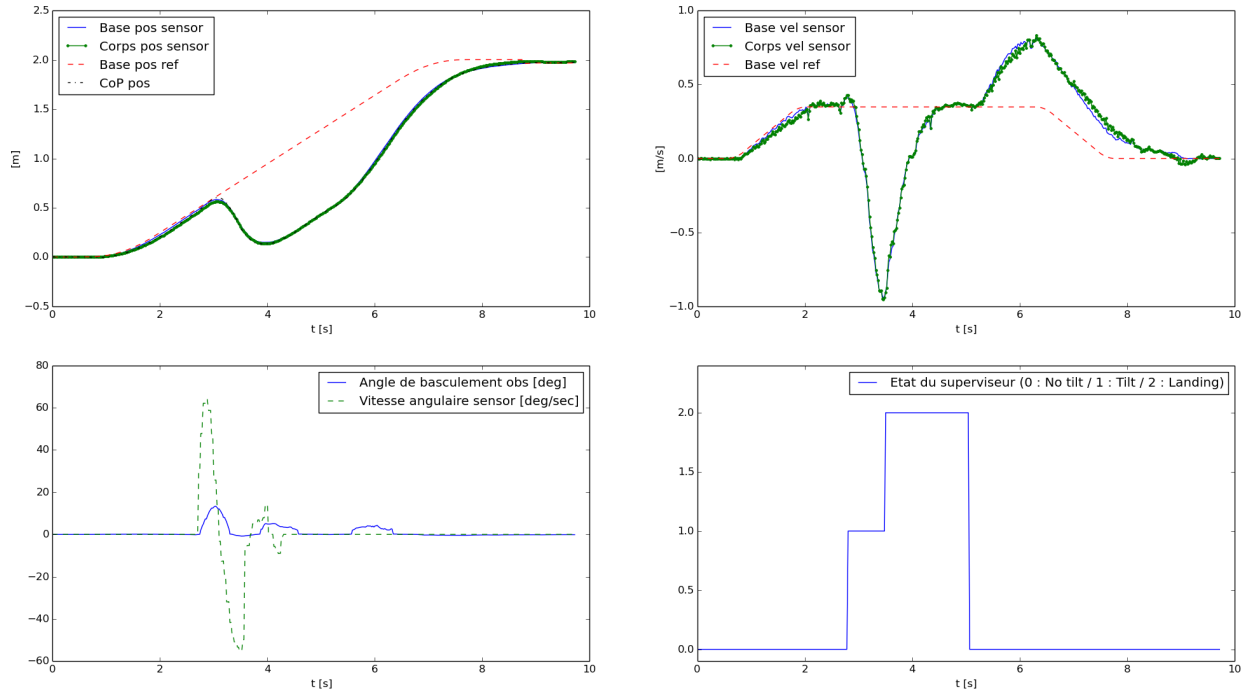


FIGURE V.11 – Compensation d'un basculement du robot en présence lorsque celui-ci est en mouvement

horizontal, puis on le pousse par les épaules afin de générer un basculement pendant qu'il est en train de se déplacer à vitesse constante.

V.3.4.2.b Analyse

Sur la figure [Fig. V.11], on peut observer que durant la période allant de 2.4s à 4.5s, le robot utilise sa base mobile pour compenser le basculement. Une fois celui-ci rattrapé, on note une forte accélération du robot afin de rattraper son retard en position.

Enfin, on peut remarquer deux rebonds après la perturbation initiale (à 4s et 5.8s). Nous avons fait en sorte que la roue en l'air lors de la perturbation ne soit pas motrice (déplacement dans l'axe allant du centre de la roue au centre du robot). Si cela n'avait pas été le cas, les rebonds auraient générés des rotations du robot autour de l'axe \vec{z} . Cette situation n'est pas prévue par la

loi de commande, et le comportement peut être très instable. La compensation de basculement dans le cas où le robot bascule est limitée à des perturbations dans la direction du déplacement uniquement.

V.3.5 Compensation de l'inclinaison du sol

V.3.5.1 Protocole expérimental

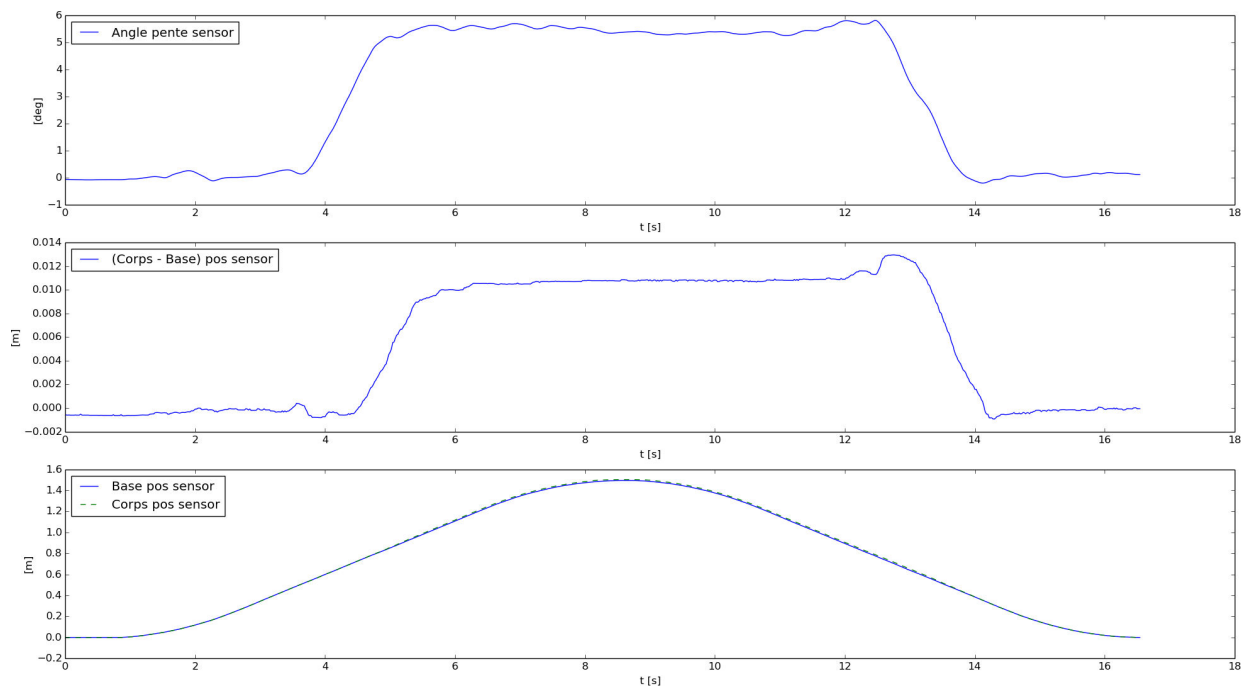


FIGURE V.12 – Compensation de l'inclinaison du sol

L'objectif de cette cinquième expérience [Fig. V.12] est de présenter le comportement de la loi de commande lorsque l'inclinaison du sol change.

Le protocole expérimental est le suivant : on dispose le robot les trois roues sur un sol plat et horizontal, puis on choisit une trajectoire de référence définie par un aller-retour de 1.5m en avant. La trajectoire de référence est dérivable deux fois. L'inclinaison du sol change à partir de 0.7m et devient une pente de 6 degrés.

V.3.5.2 Analyse

Sur la figure [Fig. V.12], on peut observer dans un premier temps que la mesure de l'angle de la pente varie de manière linéaire lors du franchissement de celle-ci. Cela est dû à la transition au cours de laquelle les roues avant du robot sont sur la pente alors que la roue arrière est sur le sol horizontal.

On peut observer également qu'afin de compenser la pente, et de conserver une CoP au centre du robot, la position du CoM du corps du robot se doit d'avancer par rapport à celui de la base mobile. Cette compensation s'opère avec un délai d'environ 0.7s. Il n'y a pas de compensation en avance de la pente car la valeur du vecteur gravité n'est pas connue dans le futur. Il y a donc nécessairement un retard entre la mesure de la pente et la compensation effective de celle-ci par le robot. Cela implique également une limitation de la vitesse du robot lors du franchissement de la pente, afin que le retard généré ne rende pas le robot instable. La vitesse du robot doit également être limitée à cause des effets dynamiques dus à la variation de la direction du vecteur gravité, qui n'ont pas été pris en compte par la loi de commande.

V.4 Conclusion

Au cours de ce chapitre, nous avons dans un premier temps présenté l'architecture de commande complète du robot. Notamment, les effets non pris en compte par la loi de commande ont été filtrés de différentes manières.

Nous avons ensuite présenté diverses expériences, mettant en avant le comportement de la loi de commande dans des situations pertinentes :

La première expérience a consisté à montrer les capacités d'adaptation de la loi de commande à des trajectoires de références non-physiquement réalisables. Notamment, on a vu que le choix des pondérations est crucial quant-au comportement du robot.

La seconde expérience a permis de montrer les performances de rejet de perturbations de la base mobile.

La troisième expérience a permis de montrer les performances et limitations de la loi de commande à compenser le basculement du robot lorsque celui-ci est initialement à l'arrêt. On peut noter que la dynamique de basculement étant très rapide, il est nécessaire de fournir des accé-

lérations très importantes de la base mobile afin de compenser des perturbations modérées. Les performances de compensation du basculement sont principalement limitées par les capacités en accélération des moteurs de la base mobile.

La quatrième expérience a permis de montrer les capacités de la loi de commande à compenser le basculement du robot lorsque celui-ci est initialement en mouvement. Une forte limitation des performances est la contrainte d'avoir un basculement uniquement dans la direction de déplacement, pour ne pas générer de moment de rotation autour de l'axe \vec{z} , qui n'est pas pris en compte par la loi de commande. De plus, emmener le robot à basculer dans une direction autre que celle de déplacement conduit à une discontinuité en vitesse de la base mobile.

Enfin, la cinquième expérience a permis de montrer les capacités d'adaptation de la loi de commande à un changement d'inclinaison du sol. Ces capacités sont principalement limités par la vitesse de franchissement de la pente, qui ne doit pas être trop grande afin de limiter les effets dynamiques dus au changement de direction du vecteur gravité. De plus, l'absence de prédiction de l'inclinaison du sol sur le futur de la trajectoire amène à compenser le changement de direction du vecteur gravité en retard, ce qui peut conduire à une instabilité de la loi de commande.

Chapitre VI

Synthèse

VI.1 Contributions

VI.1.1 Modélisation

Au cours de cette thèse, nous avons présenté dans un premier temps une modélisation du robot utilisant un modèle à deux masses ponctuelles. Ces masses permettent de modéliser les CoM de la base mobile et du corps du robot. Ce modèle a été simplifié en utilisant quatre hypothèses :

- Nous avons considéré que le robot ne pouvait se trouver que sur trois roues, ou basculer sur deux roues. Nous avons ignoré les cas de basculement sur une roue, ainsi que la chute libre.
- De plus, nous avons considéré que les effets dynamiques induits par la rotation du robot autour de l'axe \vec{z} sont négligeable. Le modèle ne dépend donc pas de la rotation du robot autour de son axe \vec{z} .
- Aussi, nous avons considéré une approximation du premier ordre des angles de basculement ψ_f , ψ_r et ψ_l .
- Enfin, nous avons considéré que les deux masses ponctuelles étaient à hauteur constante, et donc ne pouvaient se déplacer que sur un plan parallèle au sol.

Afin de résoudre *a priori* le problème de complémentarité posé par les contacts des roues avec le sol, nous avons choisi de scinder le modèle général en deux sous-modèles. Le premier considère que le robot possède ses trois roues en contact avec le sol. Le second considère que le robot est en

train de basculer sur deux de ses roues.

Concernant le premier sous-modèle, une équation du CoP en a été formulée. Concernant le second sous-modèle, une équation de la dynamique de basculement en a été formulée.

Enfin, afin de réaliser un modèle prédictif, les deux sous-modèles précédents ont été extrapolés en utilisant une dynamique échantillonnée polynomiale d'ordre 3. Il a été également précisé les conditions de validité de cette extrapolation dans le cas de la dynamique de basculement. Notamment, la période d'échantillonnage doit être différente d'une valeur dépendante des paramètres du modèle.

VI.1.2 Loi de commande

Nous avons ensuite présenté une loi de commande optimale, basée sur la résolution d'un problème quadratique. La loi de commande dispose d'un superviseur permettant de choisir un contrôleur adapté à chacun des deux modèles.

Le premier contrôleur permet d'assurer au robot de rester les trois roues en contact avec le sol, tout en contrôlant sa trajectoire ainsi que sa robustesse aux perturbations. Le second contrôleur permet de contrôler le robot lorsqu'il est en train de basculer sur deux roues, en contrôlant son basculement pour le ramener sur trois roues.

Le superviseur utilise les capteurs disponibles sur le robot afin de prédire l'existence ainsi que la vitesse angulaire de l'impact de la roue en l'air avec le sol.

VI.1.3 Résultats

Nous avons enfin présenté diverses expériences sur un robot disposant d'une implémentation de la loi de commande. Chacune de ces expériences a pour but de présenter un aspect important de la loi de commande.

Le premier essai a consisté en un suivi d'une trajectoire de référence non physiquement réalisable. Nous avons pu montrer que le choix des valeurs des pondérations associées à chaque objectif des contrôleurs influait de manière significative sur le comportement du robot. Lorsque les pondérations priorisent le suivi de trajectoire, alors celui-ci est optimal, mais la robustesse aux perturbations est médiocre. A l'inverse, en priorisant la robustesse aux perturbations, alors le suivi de trajectoire devient médiocre. Nous avons pu également constater l'effet prédictif de la loi de

commande, en observant le robot compenser en avance la partie non-réalisable de la trajectoire de référence.

Le second essai a consisté en un suivi d'une trajectoire de référence réalisable, lorsque la base mobile est soumise à une perturbation. Nous avons pu constater les capacités de la loi de commande à compenser cette perturbation. Il est intéressant de noter que le choix relatif des pondérations entre le suivi en position et le suivi en vitesse détermine le temps de compensation de la perturbation.

Le troisième essai a consisté en la compensation du basculement du robot provoqué par une perturbation, lorsque celui-ci est initialement à l'arrêt. Nous avons pu constater que le superviseur ne déclenche la compensation active de la perturbation que si l'estimation de la vitesse d'impact de la roue en l'air est supérieure à une certaine limite. Aussi, les performances de la compensation du basculement sont fortement limitées par les limites en accélération et en vitesse de la base mobile. Une perturbation moyenne amène très rapidement les commandes des roues à saturer. Les performances sont donc dans un premier temps dégradées par l'augmentation de la vitesse d'impact, et enfin par l'absence de possibilité de ramener le robot sur ses trois roues, ce qui provoque la chute du robot.

Le quatrième essai a consisté en la compensation du basculement du robot provoqué par une perturbation, lorsque celui-ci est initialement en mouvement. Nous avons pu montrer ici les limites en terme de modélisation de la dynamique de basculement. Notamment le fait de n'avoir pas considéré de rotation selon l'axe \vec{z} nous oblige à considérer des perturbation uniquement dans la direction du déplacement.

Le dernier essai a consisté en la compensation de variations de l'inclinaison du sol au cours d'un déplacement. Nous avons pu observer que l'absence de mesure prédictive du sens du vecteur gravité g occasionne un retard dans la compensation effective de la pente. Cela limite donc la vitesse de franchissement de celle-ci, afin de ne pas générer de mouvement instables, par le retard induit de la mesure de g . Aussi, nous n'avons pas pris en compte les effets dynamiques dues à un changement de sens de g , ce qui limite également la vitesse de franchissement de la pente.

VI.2 Perspectives

Plusieurs perspectives d'amélioration sont envisagées dans le futur.

Dans un premier temps, générer un moment de rotation en utilisant les bras du robot peut

grandement améliorer les performances de la compensation du basculement. Pour se faire, une solution envisagée est d'utiliser non pas un modèle à deux masses, mais plutôt un modèle à quatre masses, les deux dernières représentant chacun l'kes CoM de chaque bras.

Ensuite, afin d'accroître le potentiel d'action du robot, il peut être intéressant de ne pas limiter le corps du robot à se déplacer sur un plan parallèle au sol. Cela implique cependant de résoudre un problème non-linéaire, ce qui peut augmenter le temps de calcul.

Aussi, il est possible de représenter les différentes contraintes du système de manière plus précises. Un avantage est d'améliorer les possibilités de mouvements du robot. L'inconvénient direct est la nécessité de recalculer les matrices et vecteurs des contraintes à chaque pas de temps, ce qui augmente nécessairement le temps de calcul.

Enfin, nous avons présenté succinctement en section III.5 une approche unifiée permettant de contrôler le robot quelque soit son nombre de contact avec le sol, en utilisant un seul contrôleur et sans utiliser de superviseur.

Bibliographie

- [1] C. Ament and M. Eichhorn. Model-based control design of the compliant robot arm airarm. In *IEEE International Conference on Control Applications*, pages 334–337, 2010.
- [2] Yasin Awais, Huang Wiang, Xu Qian, and Sultan M. Saad. Humanoid robot push recovery through foot placement. In *IEEE International Conference on Mechatronics and Automation*, 2012.
- [3] N. Bascarevic, K. Jovanovic, P. Milosavljevic, V. Potkonjak, and O. Holland. Tip-over stability examination of a compliant anthropomorphic mobile robot. In *IEEE International Conference on Control Applications*, pages 1584–1589, 2012.
- [4] Stephens Benjamin J. and Atkeson Christopher G. Push recovery by stepping for humanoid robots with force controlled joints. In *IEEE International Conference on Humanoid Robots*, 2010.
- [5] J. Borenstein. Control and kinematic design of multi-degree-of freedom mobile robots with compliant linkage. In *IEEE Transactions on Robotics and Automation*, pages 21–35, 1995.
- [6] Hang C. C. The smith predictor. In *Control of Dead-time Processes*, pages 131–163, 2007.
- [7] J.C. Chiou, Lin Chun-Ying, Chen Chih-Liang, and Chien Chin-Pin. Tilting motion control in narrow tilting vehicle using double-loop pid controller. In *Asian Control Conference*, pages 913–918, 2009.
- [8] H. Dallali, G. Medrano-Cerda, N. Kashiri, N. Tsagarakis, and D. Caldwell. Decentralized feedback design for a compliant robot arm. In *European Modelling Symposium*, pages 334–337, 2010.
- [9] Hoppe Donald H.W. Quadratic programming. In *Optimization Theory*, page chapter 3, 2006.

- [10] Tao Feng, Tao Liu, Xu Wang, Zhao Xu, Meng Zhang, and Sheng-Chao Han. Modeling and implementation of two-wheel self-balancing robot equipped with supporting arms. In *IEEE/RSJ International Conference on Industrial Electronics and Applications*, pages 713–718, 2011.
- [11] H. Furuichi, Huand Jian, T. Matsuno, and T. Fukuda. Dynamic model of three wheeled narrow tilting vehicle and corresponding experiment verification. In *International Conference on Intelligent Robots and Systems*, pages 3728–3733, 2012.
- [12] Ferreau Hans Joachim, Kirches Christian, Potschka Andreas, Bock Hans Georg, and Diehl Mortis. qpOASES : a parametric active-set algorithm for quadratic programming. In *Mathematical Programming Computation*, pages 327–363, 2014.
- [13] A. Herdt, N. Perrin, and P.-B. Wieber. Walking without thinking about it. In *International Conference on Intelligent Robots and Systems*, pages 190–195, 2010.
- [14] S. Hyon. A motor control strategy with virtual musculoskeletal systems for compliant anthropomorphic robots. In *IEEE/ASME International Transactions on Mechatronics*, pages 677–688, 2009.
- [15] F. Iida, J. Rummel, and A. Seyfarth. Bipedal walking and running with compliant legs. In *IEEE International Conference on Robotics and Automation*, pages 3970–3975, 2007.
- [16] Hongzhe Jin, Jongmyung Hwang, and Jangmyung Lee. A balancing control strategy for a one-wheel pendulum robot based on dynamic model decomposition : Simulations and experiments. In *IEEE/ASME Transactions on Mechatronics*, 2011.
- [17] Wang Jiuguang. Humanoid push recovery with robust convex synthesis. In *IEEE International Conference on Intelligent Robots and Systems*, 2012.
- [18] Rebula Joh, Canas Fabian, Pratt Jerry, and Goswami Ambarish. Learning capture points for humanoid push recovery. In *IEEE International Conference on Humanoid Robots*, 2007.
- [19] Lafaye Jory, Collette Cyrille, and Pierre-Brice Wieber. Model predictive control for tilt recovery of an omnidirectional wheeled humanoid robot. In *International Conference on Robotics and Automation*, 2015.
- [20] Lafaye Jory, Gouaillier David, and Pierre-Brice Wieber. Linear model predictive control of the locomotion of pepper, a humanoid robot with omnidirectional wheels. In *International Conference on Humanoid Robots*, pages 336–341, 2014.

- [21] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of Zero Moment Point. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1620–1626, september 2003.
- [22] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode : A simple modeling for a biped walking pattern generation. In *International Conference on Intelligent Robots and Systems*, pages 239–246, 2001.
- [23] Kiattisin Kanjanawanishkul and Andreas Zell. Path following for an omnidirectional mobile robot based on model predictive control. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 3341–3346, Kobe, Japan, May 12-17 2009.
- [24] Alipour Khalil and Ali A. Moosavian. Point-to-point stable motion planning of wheeled mobile robots with multiple arms for heavy object manipulation. In *IEEE International Conference on Robotics and Automation*, 2011.
- [25] Jaeoh Lee, Seongik Han, and Jangmyung Lee. Decoupled dynamic control for pitch and roll axes of the unicycle robot. In *IEEE/ASME Transactions on Industrial Electronics*, 2015.
- [26] Yanjie Li, Dalong Tan, Zhenwei Wu, and Hua Zhong. Dynamic stability analyses based on zmp of a wheel-based humanoid robot. In *International Conference on Robotics and Biomimetics*, Kunming, China, 2006.
- [27] Zhibin Li, N.G. Tsagarakis, and D.G. Caldwell. A passivity based admittance control for stabilizing the compliant humanoid coman. In *IEEE International Conference on Humanoid Robots*, pages 43–49, 2012.
- [28] Bingo Wing-Kuen Ling, Charlotte Yuk-Fan Ho, Qingyun Dai, and Joshua D. Reiss. Reduction of quantization noise via periodic code for oversampled input signals and the corresponding optimal code design. In *Digital Signal Processing*, 2013.
- [29] Naveau M., Carpentier J., Barthelemy S., Stasse O., and Soueres P. Metapod - template meta-programming applied to dynamics : Cop-com trajectories filtering. In *IEEE International Conference on Humanoid Robot*, 2014.

- [30] Sebastian O.H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. 2010. "https://www.samba.org/tridge/UAV/madgwick_internal_report.pdf".
- [31] M.A. Minor, B.W. Albiston, and C.L. Schwensen. Simplified motion control of a two-axle compliant framed wheeled mobile robot. In *IEEE Transactions on Robotics*, pages 491–506, 2006.
- [32] Morisawa Mitsuharu, Harada Kensuke, Kajita Shuuji, Nakaoka Shinichira, Fujiwara Kiyoshi, Kanehiro Fumio, Kaneko Kenji, and Hirukawa Hirohisa. Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution. In *IEEE International Conference on Robotics and Automation*, 2007.
- [33] N. Oda, K. Kushida, and M. Yamazaki. Vision-based control evaluation with estimating foot sole floating angle for biped walking robot. In *International Workshop on Advanced Motion Control*, pages 344–349, 2014.
- [34] Wieber P. B. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE International Conference on Humanoid Robot*, 2006.
- [35] Gill P. E., Hammarling S. J., Murray W., Saunders M. A., and Wright M. H. User’s guide for lssol (version 1.0) : A fortran package for constrained linear least squares and convex quadratic programming. In *Systems Optimization Laboratory, Department of Operations Research, Stanford University*, 1986.
- [36] Kaustubh Pathak, Jaume Franch, and Sunil Kumar Agrawal. Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *IEEE Transactions on Robotics*, 21(3) :505–513, 2005.
- [37] Marker R. Timothy and Arora Jasbir S. The weighted sum method for multi-objective optimization : new insights. In *Structural and Multidisciplinary Optimization*, pages 853–862, 2010.
- [38] Xiaogang Ruan, Jing Chen, Jianxian Cai, and Lizhen Dai. Balancing control of two-wheeled upstanding robot using adaptive fuzzy control method. In *IEEE/RSJ International Conference on Intelligent Computing and Intelligent Systems*, pages 95–98, 2009.

- [39] P. Sardain and G. Bessonnet. Forces acting on a biped robot. Center of Pressure—Zero Moment Point. *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 34(5) :630–637, september 2004.
- [40] Yi Seung-Joon, Zhang Byoung-Tak, Hong Dennis, and Lee Daniel D. Online learning of a full body push recovery controller for omnidirectional walking. In *IEEE International Conference on Humanoid Robots*, 2011.
- [41] W. Svensson and U. Holmberg. Ground angle estimator from an ankle foot orthosis based on strain sensing and fourier series. In *IEEE International Conference on Mechatronics and Automation*, pages 203–206, 2008.
- [42] N. Takasuka, T. Tanaka, S. Kaneko, T. Tada, and S. Suzuki. Self-localization of a mobile robot on irregular ground using a laser range finder. In *World Automation Congress*, pages 1–6, 2010.
- [43] J. Togashi, T. Matsuda, and K. Mitobe. A low cost and lightweight wire driven robot arm by using elastic strings. In *IEEE International Symposium on System Integration*, pages 436–440, 2014.
- [44] Xin Wang, Mantian Li, Pengfei Wang, and Lining Sun. Running and turning control of a quadruped robot with compliant legs in bounding gait. In *IEEE International Conference on Robotics and Automation*, pages 511–518, 2011.
- [45] K. Watanabe. Control of an omnidirectional mobile robot. In *KES (I)*, pages 51–60. IEEE, 1998.
- [46] Mao Wentao, Qin Guihe, and Lee Ju-Jang. Humanoid push recovery strategy for unknown input forces. In *IEEE International Conference on Mechatronics and Automation*, 2009.
- [47] M. Yamazaki and N. Oda. Foot adaptation to obstacle with vision-based landing angle estimation for biped robot. In *Annual Conference of the IEEE Industrial Electronics Society*, pages 5254–5259, 2014.
- [48] Jie Zhao, Mei Xiong, and Hongzhe Jin. Dynamics and a convenient control design approach for a unicycle robot. In *IEEE International Conference on Information and Automation*, 2010.

Annexe A

Pepper, un robot humanoïde à roues omnidirectionnelles

Pepper [Fig. A.1] est un robot humanoïde conçu par Aldebaran pour Soft Bank Mobile. Il s'agit d'un robot social qui a pour but d'accueillir et de divertir. Il peut également être utilisé en tant que guide dans un magasin par exemple. Softbank a pour objectif d'en faire un robot-compagnon accessible au grand public et étant destiné à interagir dans les maisons.

Il fait 1m20 de haut et pèse 28kg.

Afin de se déplacer, il dispose d'une base mobile composée de trois roues omnidirectionnelles. Chaque roue étant actionnées de façon symétrique, le déplacement de la base mobile est également omnidirectionnel.

Afin de changer sa posture, Pepper dispose de 17 articulations [Fig. A.2]. trois lui permettent de contrôler la position de la jambe, deux pour la tête, et six pour chaque bras.

Pour observer son environnement, il dispose d'une caméra 3D dont l'émetteur et le récepteur sont disposés dans ses yeux. Il possède également deux caméras 2D sur le front et dans la bouche. La base mobile dispose de trois bumpers recouvrant les roues, de deux sonars, ainsi que de trois capteurs lasers.

L'odométrie du robot est obtenue par des capteurs magnéto-résistifs disposés sur chaque articulation, ainsi que d'une centrale inertielle dans la base mobile.

Enfin, pour interagir avec les personnes, il dispose de microphones sur la tête, de deux haut-



FIGURE A.1 – Présentation de Pepper, robot humanoïde conçu par Aldebaran

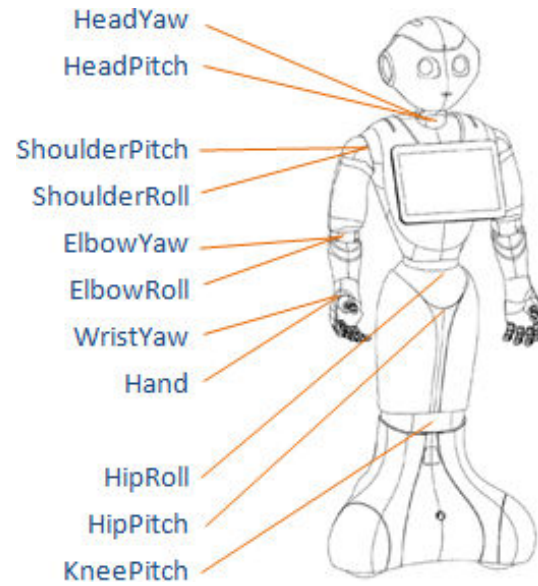


FIGURE A.2 – Présentation des articulations de Pepper

parleurs, de leds disposées dans le yeux et les oreilles, ainsi que d'une tablette tactile Android.

Annexe B

***MPC-WalkGen*, librairie C++ implémentant la commande par modèle prédictif**

MPC-WalkGen est une librairie développée dans le cadre de cette thèse, et ayant pour but d'implémenter la loi de commande présentée. La librairie est codée en C++, et utilise le solveur de programme quadratique *QPOases*.

La librairie est open-source, et est accessible sur github à l'adresse suivante :

<https://github.com/aldebaran/mpc-walkgen>

MPC-WalkGen est codée de façon modulaire vis à vis du choix du solveur ainsi que du problème à résoudre. La loi de commande est séparée en trois modules :

- Le module *model*, implémentant les différents modèles dynamiques. Pour chaque modèle, il est possible d'en obtenir les différents matrices et vecteurs représentant sa dynamique prédictive.
- Le module *function*, implémentant les fonctions de contrainte et d'objectif appliqués aux modèles. Pour chaque fonction, il est possible d'en obtenir sa valeur, son gradient et sa hessienne en un point de fonctionnement.
- Le module *solver*, implémentant un solveur de programme quadratique choisi.

Les modules *model* et *function* permettent de construire les matrices et vecteurs du problème à résoudre, le résultat étant envoyé au module *solver* sélectionné.

Enfin, cette librairie a été optimisée afin d'être efficace en terme de temps de calcul. De façon

non exhaustive :

- Les matrices n'évoluant pas au cours de l'expérience sont calculés hors ligne
- L'utilisation de la librairie Eigen permet de profiter d'optimisations hardware quant-aux calculs matriciels
- Le problème est construit de manière à pouvoir utiliser les méthodes de type *warm-start* propres aux solveurs de programmes quadratiques